

10/29/01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Osamu HORI, et al.

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HEREWITH

FOR: METHOD OF DESCRIBING OBJECT REGION DATA, APPARATUS FOR GENERATING OBJECT REGION DATA, VIDEO PROCESSING METHOD, AND VIDEO PROCESSING APPARATUS

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §119(e).
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

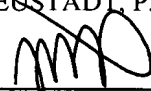
<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
Japan	11-221424	August 4, 1999

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
(B) Application Serial No.(s)
 - ☐ are submitted herewith
 - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak
Registration No. 24,913



22850

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

1c893 U.S. PTO
09/633231
08/04/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

1999年 8月 4日

出 願 番 号
Application Number:

平成11年特許願第221424号

出 願 人
Applicant(s):

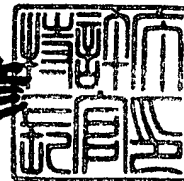
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 6月 9日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3044220

【書類名】 特許願

【整理番号】 A009904317

【提出日】 平成11年 8月 4日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/00

【発明の名称】 物体領域情報記述方法及び物体領域情報生成装置並びに
映像情報処理方法及び情報処理装置

【請求項の数】 18

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝研
究開発センター内

【氏名】 堀 修

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝研
究開発センター内

【氏名】 金子 敏充

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝研
究開発センター内

【氏名】 三田 雄志

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝研
究開発センター内

【氏名】 山本 晃司

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プールの可否】 要

【書類名】 明細書

【発明の名称】 物体領域情報記述方法及び物体領域情報生成装置並びに映像情報処理方法及び情報処理装置

【特許請求の範囲】

【請求項 1】

映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、

映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、

前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記対象となる物体の領域に関する情報を記述することを特徴とする物体領域情報記述方法。

【請求項 2】

映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、

映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、

前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似し、

前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報とを用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする物体領域情報記述方法。

【請求項 3】

前記参照物体領域を、ビットマップで表現することを特徴とする請求項 1 または 2 に記載の物体領域情報記述方法。

【請求項 4】

前記参照物体領域を、該物体領域に対する近似図形の代表点で表現することを特徴とする請求項 1 または 2 に記載の物体領域情報記述方法。

【請求項 5】

前記近似図形の種類に応じて、前記領域変換および前記変換パラメータの種類を選択することを特徴とする請求項 4 に記載の物体領域情報記述方法。

【請求項 6】

前記変換パラメータを求めるにあたって、該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が最小となるように、前記参照物体領域を選択することを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 7】

前記参照物体領域として、前記対象となる物体の領域が存在する連続した複数フレームにおける中間フレームの当該対象となる物体の領域を選択することを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 8】

あるフレームにおける前記対象となる物体の領域を前記参照物体領域として、該参照物体領域をもとにして、該あるフレームに後続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求めるとともに、

該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が予め定められたしきい値を上回った場合には、新たなフレームにおける前記対象となる物体の領域を新たな参照物体領域として更新することを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 9】

連続する複数フレームの中央に位置するフレームにおける前記対象となる物体の領域を前記参照物体領域とし、該参照物体領域をもとにして、該中央に位置するフレームの前後に連続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求める処理を行い、

該変換パラメータで生成される予測物体領域と実際の物体の領域との誤差が予め定められたしきい値を上回る範囲に属する連続する複数フレームに対して、前記処理を再帰的に行うことを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 10】

対象となる物体の領域を、複数の領域に分割し、

この分割によって得られた前記複数の領域の各々を独立して前記変換パラメータで表現することによって、前記対象となる物体の領域に関する情報を記述することを特徴とする請求項 1 ないし 9 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 1 1】

前記物体に関連付けられている関連情報または該関連情報へのアクセス方法を示す情報を併せて記述することを特徴とする請求項 1 ないし 1 0 のいずれか 1 項に記載の物体領域情報記述方法。

【請求項 1 2】

映像データをもとにして請求項 1 ないし 1 1 のいずれか 1 項に記載の物体領域情報記述方法により記述された 1 または複数の物体の領域に関する情報を含む物体領域データを格納した記録媒体。

【請求項 1 3】

連続する複数のフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための映像情報処理方法であって、

映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関する情報および該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情報を取得し、

前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に変換し、

この変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定することを特徴とする映像情報処理方法。

【請求項 1 4】

前記参照物体領域が代表点で表現されている場合には、該参照物体領域をビットマップに展開した後に、前記変換により求められた位置が、該ビットマップに展開された参照物体領域の内部に存在するか否かを調べることを特徴とする請

求項 13 の映像情報処理方法。

【請求項 15】

前記物体が指定されたと判定された場合には、前記物体に関連付けられている関連情報を呈示することを特徴とする請求項 13 または 14 に記載の映像情報処理方法。

【請求項 16】

映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であって、

映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、

前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記物体の領域に関する情報を表現する手段とを備えたことを特徴とする物体領域情報生成装置。

【請求項 17】

映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であって、

映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、

前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似する手段とを備え、

前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報とを用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする物体領域情報生成装置。

【請求項 18】

連続するフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための情報処理装置であって、

映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関する情報および該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情

報を取得する手段と、

前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に逆変換する手段と、

この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定する手段とを備えたことを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、映像データ中の物体の領域に関する情報を記述するための物体領域情報記述方法、映像データ中の物体の領域に関する情報を生成するための物体領域情報生成装置、並びに映像データ中の物体に対する指定を受け所定の処理を行う情報処理装置及びそのための映像情報処理方法に関する。

【0002】

【従来の技術】

映像データ中の物体領域を記述する方法として、従来はMPEG-4のオブジェクトコーディングが知られている。このMPEG-4のオブジェクトコーディングはフレーム毎にオブジェクト領域をビットマップとして表現し、さらに、フレーム内およびフレーム間で情報圧縮を行っている。形状およびテクスチャーは別々にコーディングされる。

【0003】

しかし、このオブジェクトコーディングでは、任意形状でかつテクスチャーをもった物体領域をコンパクトに表現することを目的としているため、コーディングされたオブジェクトは、フレーム間圧縮されており、フレーム毎に物体領域の情報を取り出すことが困難であった。また、映像中の領域の位置情報のみを表現したい場合には、オーバースペックでありCodecに処理時間がかかるという問題があった。

【0004】

【発明が解決しようとする課題】

上述のようにMPEG-4のCodecを用いると、不必要な情報をコーディングするため、無駄が多いばかりでなく、フレーム間圧縮を行っているため、フレーム毎の情報を取り出すことが困難であるという問題があった。

【0005】

本発明は、上記事情を考慮してなされたもので、映像中の所望の物体の領域を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にする物体領域情報記述方法及び物体領域情報生成装置を提供することを目的とする。

【0006】

また、本発明は、ユーザによる映像データ中の物体の指示の判定を容易にする物体領域情報記述方法及び物体領域情報生成装置並びに映像情報処理方法及び情報処理装置を提供することを目的とする。

【0007】

【課題を解決するための手段】

本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0008】

また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似し、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報とを用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0009】

好ましくは、前記変換パラメータをそのまま記述することによって、前記対象となる物体の領域に関する情報を記述するようにしてもよい。

【0 0 1 0】

好ましくは、前記参照物体領域を、ビットマップで表現するようにしてもよい。

【0 0 1 1】

好ましくは、前記参照物体領域を、該物体領域に対する近似図形の代表点で表現するようにしてもよい。

【0 0 1 2】

好ましくは、前記近似図形は、傾きのない矩形もしくは楕円、傾きのある矩形もしくは楕円、または多角形であるようにしてもよい。

【0 0 1 3】

好ましくは、前記近似図形の種類に応じて、前記領域変換および前記変換パラメータの種類を選択するようにしてもよい。

【0 0 1 4】

好ましくは、前記近似図形の代表点は予め順序付けられたものであるようにしてもよい。

【0 0 1 5】

好ましくは、前記変換パラメータを求めるにあたって、該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が最小となるように、前記参照物体領域を選択するようにしてもよい。

【0 0 1 6】

好ましくは、前記参照物体領域として、前記対象となる物体の領域が存在する連続した複数フレームにおける中間フレームの当該対象となる物体の領域を選択するようにしてもよい。

【0 0 1 7】

好ましくは、あるフレームにおける前記対象となる物体の領域を前記参照物体領域として、該参照物体領域をもとにして、該あるフレームに後続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求めるとと

もに、該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が予め定められたしきい値を上回った場合には、新たなフレームにおける前記対象となる物体の領域を新たな参照物体領域として更新するようにしてもよい。

【0018】

好ましくは、連続する複数フレームの中央に位置するフレームにおける前記対象となる物体の領域を前記参照物体領域とし、該参照物体領域をもとにして、該中央に位置するフレームの前後に連続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求める処理を行い、該変換パラメータで生成される予測物体領域と実際の物体の領域との誤差が予め定められたしきい値を上回る範囲に属する連続する複数フレームに対して、前記処理を再帰的に行うようにしてもよい。

【0019】

好ましくは、対象となる物体の領域を、複数の領域に分割し、この分割によって得られた前記複数の領域の各々を独立して前記変換パラメータで表現することによって、前記対象となる物体の領域に関する情報を記述するようにしてもよい。

【0020】

好ましくは、前記物体に関連付けられている関連情報または該関連情報へのアクセス方法を示す情報を併せて記述するようにしてもよい。

【0021】

また、本発明は、映像データをもとにして請求項 1 ないし 11 のいずれか 1 項に記載の物体領域情報記述方法により記述された 1 または複数の物体の領域に関する情報を含む物体領域データを格納した記録媒体である。

【0022】

また、本発明は、連続する複数のフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための映像情報処理方法であって、映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関

する情報および該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情報を取得し、前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に逆変換し、この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定することを特徴とする。

【0023】

好ましくは、前記参照物体領域が代表点で表現されている場合には、該参照物体領域をビットマップに展開した後に、前記逆変換により求められた位置が、該ビットマップに展開された参照物体領域の内部に存在するか否かを調べるようにしてもよい。

【0024】

好ましくは、前記物体が指定されたと判定された場合には、前記物体に関連付けられている関連情報を呈示するようにしてもよい。

【0025】

また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記物体の領域に関する情報を表現する手段とを備えたことを特徴とする。

【0026】

また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似する手段とを備え、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報とを用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【 0 0 2 7 】

また、本発明は、連続するフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための情報処理装置であって、映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関する情報および該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情報を取得する手段と、前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に逆変換する手段と、この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定する手段とを備えたことを特徴とする。

【 0 0 2 8 】

なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

【 0 0 2 9 】

また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムを記録したコンピュータ読取り可能な記録媒体としても成立する。

【 0 0 3 0 】

本実施形態では、例えば、連続または断続にフレーム中に存在する領域と表現したい物体の領域情報をフレーム毎に、近似図形またはビットマップで記述し、物体領域の存在する区間において、少なくともひとつの参照となる物体領域を選択する。その参照物体領域から他の物体領域への変換パラメータを計算し、変換パラメータを連続するフレーム順に並べて、変換パラメータの軌跡を描く。あるいは、さらに、この軌跡を、近似関数を用いて少ない情報量で表現を行う。この方法によって、時空間情報を持った物体領域は、例えば、参照物体領域と変換パラメータの軌跡を近似した関数と物体領域の存在した区間の情報で表現される。

また、例えば、任意のフレームの物体領域の情報を取り出すためには、近似関数から変換パラメータを求め、参照物体領域から、その変換パラメータを用いて物体領域を変形され予測物体領域を求め、その領域を求める物体領域とする。

また、例えば、マウスポインタで指定された位置が時空間物体領域の中か外にあるか判定するには、求められた変換パラメータを用いて逆変換を行いマウスポインタの位置情報を参照物体領域のあるフレームの座標に変換し、参照物体領域の内か外かを判定し、指定されたフレームの物体領域の中にあるかを判定する。このように、容易に物体領域の内外判定ができるため、操作性の良いハイパーメディア・アプリケーションを実現できる。

【 0 0 3 1 】

このように、本発明によれば、映像中を移動し、形が変化する物体領域の形および位置情報を少ない容量で記述することができ、フレーム毎の物体領域の情報を高速に取り出すことができる。特に、ハイパーメディア・アプリケーションのように、マウスポインターを用いて、映像中の物体を指定した場合、マウスポインターの位置が物体の領域の中に含まれているか含まれていないかを高速に判定し、インタラクティブ性の高い操作環境を提供することができる。また、物体領域の動き、大きさなどを容易に知ることができ、検索システムや監視するシステムなどのアプリケーションに利用できる。

【 0 0 3 2 】

【発明の実施の形態】

以下、図面を参照しながら発明の実施の形態を説明する。

【 0 0 3 3 】

（第 1 の実施形態）

図 1 に、本発明の第 1 の実施形態に係る物体領域データ変換装置の構成例を示す。図 1 に示されるように、本物体領域データ変換装置は、映像データ記憶部 2、物体領域処理部 4、変換パラメータ処理部 6、関数近似処理部 8、物体領域データ記憶部 10 を備えている。なお、本処理においてユーザの操作を介入させる形態を取る場合には、映像（動画像）データを例えばフレーム単位で表示させ、ユーザの指示入力等を受け付けるグラフィカル・ユーザ・インタフェース（GU

I) が用いられる (図 1 では省略している)。

【0034】

物体 (オブジェクト) 領域は、映像中における一纏まりの領域部分であり、例えば、人、動物、植物、車、建物、道、川、太陽、雲など (あるいはその一部分、例えば人の頭、車のボンネット、建物の玄関など)、オブジェクトとして把握し得るものならどのようなものでも扱うことができる。

【0035】

映像データ記憶部 2 は、映像データが記憶されているもので、例えばハードディスクや光ディスク、半導体メモリなどで構成される。

【0036】

物体領域処理部 4 は、基準となるフレームの物体領域 (参照物体領域) や対象となるフレームの物体領域 (対象物体領域) を得るための処理を行う。

【0037】

変換パラメータ処理部 6 は、参照物体領域をもとにして、対象物体領域の変換パラメータを算出する処理を行う。

【0038】

関数近似処理部 8 は、物体領域の変換パラメータの各々について、その時系列的な軌跡を時間関数に変換する処理を行う。なお、後述するように、変換パラメータ自体を記述する場合には、この関数近似処理部 8 は不要となる。

【0039】

物体領域データ記憶部 10 は、変換パラメータの各々について、その時系列的な軌跡を近似した関数式を表現するデータを含む物体領域データが記憶される記憶媒体である。

【0040】

なお、本実施形態では、参照物体領域の更新判定処理を行うが、その部分は図 1 では省略している。

【0041】

映像データ記憶部 2 および領域データ記憶部 10 は、別々の記憶装置によって構成されていてもよいが、それらの全部または一部が同一の記憶装置によって構

成されていてもよい。

【0042】

また、本物体領域データ変換装置は、計算機上でソフトウェアを実行する形で実現することもできる。

【0043】

図2に、本実施形態の物体領域データ変換装置の処理手順の一例を示す。

【0044】

まず、ステップS101では、映像中の物体領域の初期値を入力する処理を行う。

【0045】

この処理は、画像処理により自動的に行われても良いし、また、GUIを介して手動で行われても良い。例えば、手動で行う場合には、処理対象となる映像中の物体の輪郭をマウスやタッチパネル等のポインティング・デバイスにより指定する。手動で入力された物体の輪郭内部を物体領域の初期値としてもよいし、また、入力された輪郭をSnakesとよばれる動的輪郭モデル（例えばM. Kass, A. Witkin and D. Terzopoulos著、「Snakes: Active contour models」、Proceedings of the 1st International Conference on Computer Vision, pp. 259-268, 1987年を参照）を用いた手法により画像中の物体の輪郭線にフィッティングさせ、フィッティング後の輪郭の内部を物体領域の初期値としても良い。

【0046】

この物体領域の初期値は、参照物体領域として登録しておく。登録方法としては、物体の領域内には1を、領域外には0をそれぞれ対応させた2値のビットマップを作成し、記憶しておく方法がある。

【0047】

また、参照物体領域の含まれるフレームを、参照フレームとして登録しておく。

【0048】

次に、ステップ S 1 0 2 では、処理対象となっている物体領域（以下、対象物体領域と呼ぶ）の変換パラメータを算出する処理を行う。

【 0 0 4 9 】

この処理は、例えば、対象物体領域内のオプティカルフロー算出処理とオプティカルフローからの変換パラメータ算出処理との 2 つの処理の組み合わせにより実現することができる。対象物体領域内のオプティカルフロー算出処理は、参照フレームから現フレームにかけての物体領域内の各画素（もしくは数画素からなるブロック）の動きを算出する処理である。

【 0 0 5 0 】

図 3 に、各フレームにおいて物体領域内のオプティカルフローを求める処理例の概要を示す。

【 0 0 5 1 】

図 3 において、2 0 1 は参照フレーム、2 0 2 は参照フレームの次のフレーム、そして 2 0 3 はフレーム 2 0 2 の次のフレームを表している。2 0 4, 2 0 5, 2 0 6 はそれぞれのフレームにおける物体の領域である。2 0 7 はフレーム 2 0 1 からフレーム 2 0 2 への物体領域のオプティカルフローである。また、2 0 8 はフレーム 2 0 1 からフレーム 2 0 3 への物体領域のオプティカルフローである。

【 0 0 5 2 】

このようにオプティカルフローを求める方法は、参照物体領域と任意のフレームにおける物体領域を直接関係させることができるため、任意フレームの物体領域を算出したり、指定された座標が物体の内部か外部かを判定したりする処理が容易である。

【 0 0 5 3 】

図 3 の例では参照フレームから現フレームにかけての物体領域内の各画素（もしくは数画素からなるブロック）のオプティカルフローを求めたが、その代わりに、一つ前のフレームから現フレームにかけての物体領域内の各画素（もしくは数画素からなるブロック）のオプティカルフローを求めるようにしても良い。図 4 に、この場合の処理例の概要を示す。

【0054】

図4において、301は参照フレーム、302は参照フレームの次のフレーム、そして303はフレーム302の次のフレームを表している。304、305、306はそれぞれのフレームにおける物体の領域である。307はフレーム201からフレーム202への物体領域のオプティカルフローである。また、308はフレーム302からフレーム303への物体領域のオプティカルフローである。

【0055】

このようなオプティカルフローの算出方法は、図3の方法に比べてパラメータの変動が少なくなるが、任意フレームの物体領域の算出が面倒になる。図3と図4のどちらの方法を用いても構わないが、本実施形態では、図3の方法でオプティカルフローを算出するものとして説明を続ける。

【0056】

なお、オプティカルフローを求める具体的な方法は多数提案されており（例えば、J. L. Barron, D. J. Fleet and S. S. Beauchemin著、「Performance of Optical Flow Techniques」、International Journal of Computer Vision, vol. 12, no. 1, pp. 43-77, 1994年を参照）、どのような方法でも用いることが可能である。

【0057】

また、参照物体領域内の特徴点を複数選択し、これら特徴点を中心とするブロックをテンプレートとするテンプレートマッチングにより求めた移動ベクトルをオプティカルフローの代わりに用いても良い。

【0058】

次に、オプティカルフローから変換パラメータを算出する処理を行うが、求める変換パラメータは前提とする変換モデルにより異なる。

【0059】

本実施形態では、

パラメータ数が1のモデルとして、「拡大縮小モデル」、「回転モデル」、
パラメータ数が2のモデルとして、「平行移動モデル」、
パラメータ数が4のモデルとして、「拡大縮小・回転・平行移動モデルの複合
モデル」（ここでは4-parameter変換モデルと呼ぶ）、
パラメータ数が6のモデルとして、「affine変換モデル」、
パラメータ数が8のモデルとして、「射影変換モデル」、
パラメータ数が12のモデルとして、「parabolic変換モデル」が選
択できるものとする。

【0060】

これらのモデルを数式で表したのが数式(1)から数式(7)である。

【0061】

【数 1】

$$\begin{cases} x' = a_0 x \\ y' = a_0 y \end{cases} \quad \dots (1)$$

$$\begin{cases} x' = x \cos a_0 - y \sin a_0 \\ y' = x \sin a_0 + y \cos a_0 \end{cases} \quad \dots (2)$$

$$\begin{cases} x' = x + a_0 \\ y' = y + a_1 \end{cases} \quad \dots (3)$$

$$\begin{cases} x' = a_0 x + a_1 y + a_2 \\ y' = a_1 x - a_0 y + a_3 \end{cases} \quad \dots (4)$$

$$\begin{cases} x' = a_0 x + a_1 y + a_2 \\ y' = a_3 x + a_4 y + a_5 \end{cases} \quad \dots (5)$$

$$\begin{cases} x' = \frac{a_0 x + a_1 y + a_2}{a_3 x + a_4 y + 1} \\ y' = \frac{a_5 x + a_6 y + a_7}{a_3 x + a_4 y + 1} \end{cases} \quad \dots (6)$$

$$\begin{cases} x' = a_0 x^2 + a_1 xy + a_2 y^2 + a_3 x + a_4 y + a_5 \\ y' = a_6 x^2 + a_7 xy + a_8 y^2 + a_9 x + a_{10} y + a_{11} \end{cases} \quad \dots (7)$$

【0 0 6 2】

数式 (1) は拡大縮小モデル、数式 (2) は回転モデル、数式 (3) は平行移動モデル、数式 (4) は 4-parameter 変換モデル、数式 (5) は affine 変換モデル、数式 (6) は射影変換モデル、数式 (7) は parabo

1 i c 変換モデルに相当する。数式中、参照物体領域中の座標を (x, y) 、対象物体領域における物体の対応点の座標を (x', y') としている。各変換モデルでは、両フレームにおける対応点の関係が、それぞれの数式のように $a_0 \sim a_{11}$ のパラメータを用いて表現できることを仮定している。もちろん、上述したモデル以外のパラメトリックなモデルを用意しても構わない。

【0 0 6 3】

変換パラメータの算出には、最小二乗法を用いることができる。これは、オプティカルフローにより求められた (x, y) と (x', y') との組み合わせを変換モデルの数式に代入した際に生じる誤差の 2 乗和を最小とするように変換パラメータを決定する方法である。これは古典的な方法で、行列演算で容易に実行することができる。

【0 0 6 4】

次に、ステップ S 1 0 3 では、算出した物体領域の変換パラメータを時間関数へ変換（近似）する処理を行う。

【0 0 6 5】

すなわち、ある時間区間における n 個の変換パラメータ a_i ($0 \leq i \leq n-1$) (例えば、 $n=12$) を、

$$a_i = f_i(t) \quad (f_i(t) \text{ は時間 } t \text{ の関数})$$

と表現する。

この時間区間とは、同一の参照物体領域を用いてオプティカルフローを算出したフレームが含まれる時間区間である。また、 $f_i(t)$ は多項式、スプライン関数、定数などとすることができる。

【0 0 6 6】

図 5 に、オプティカルフローから算出された（ある 1 つの） a_i を関数により表現した様子を示す。図 5 において、4 0 1 は関数表現の対象となっている時間区間、4 0 2 はオプティカルフローから算出された a_i の値、4 0 3 は a_i を表現する関数 $a_i = f_i(t)$ を表している。

【0 0 6 7】

このように関数により表現するメリットは、記述データを少なくできることで

ある。例えば、関数として2次以下の多項式を用いるとすると、この関数は3つの実数により表現することができるため、ある時間区間全てのパラメータの値を記述するのに3つの実数で十分になる。

【0068】

パラメータを表現する関数を多項式やスプライン関数とする場合には、変換対象となっている時間区間における a_i の値と $f_i(t)$ により算出される値との誤差が小さくなるように決定する。例えば、最小二乗法を利用すれば容易に計算ができる。

【0069】

なお、この近似関数を求める処理は、例えば、当該物体領域に関する各フレームにおけるパラメータ値が得られる毎に行う（例えば各フレームのパラメータ値が得られる毎に近似を行うとともに近似誤差を求め、近似誤差が一定の範囲に収まるように近似区間を適宜分割する方法）ようにしてもよいし、参照物体領域が更新されて参照物体一定区間が定まった後に一括して行うようにしてもよい。

【0070】

このステップS103の処理手順については後でより詳細に説明する。

【0071】

次に、ステップS104では、参照物体領域の更新判定処理を行う。

【0072】

本実施形態は、任意のフレームにおける物体の領域を、参照フレームにおける参照物体領域と、参照物体領域の変換パラメータとにより表現するものである。しかし、表現したい物体領域と参照物体領域とがあまりに異なった形状である場合には、変換パラメータにより参照物体領域を移動・変形しても、表現したい物体領域に似ている形状が得られない。このような場合には、参照物体領域を別のフレームにおける物体領域に変更するのが効果的である。そこで、本実施形態では、ステップS104にて、そのような変更をするか否かを判定するようにしている。

【0073】

この判定には、例えば、実際の物体領域と予測物体領域との誤差があらかじめ

定められているしきい値を上回るかどうかにより判断する方法を用いることができる。予測物体領域とは、参照物体領域を変換パラメータにより変換した領域のことである。変換に用いる変換パラメータは、時間関数 $a_i = f_i(t)$ から算出した値を用いる。また、実際の物体領域は、予測物体領域に対して *Snakes* 等の輪郭抽出処理を施して算出する。さらに、実際の物体領域と予測物体領域との誤差としては、例えば、両領域の共通部分の面積と共通しない部分の面積との比を用いることができる。

【0074】

次に、ステップ S105 では、先のステップ S104 の判定処理において参照物体領域の更新が決定された場合に、参照物体領域更新処理を行う。この処理は、基本的にはステップ S101 の処理と同様である。すなわち、新たなフレームを参照フレームとして登録し、参照物体領域を表現する 2 値のビットマップを作成する処理である。また、参照フレームにおける物体領域を参照物体領域として登録しておく。

【0075】

次に、ステップ S106 では、映像中の物体領域の記述処理が終了かどうかを判定する。この判定基準としては、例えば、ビデオの終端かどうか、物体領域存在区間の終端かどうか、ユーザにより終了が指示されたかどうか、などがある。ステップ S102 からステップ S104 またはステップ S105 までの処理が、ステップ S106 で終了と判定されるまで繰り返し実行される。

【0076】

そして、ステップ S107 において、これまでの処理により計算された物体領域の記述情報（物体領域データ）を、予め定められた記述フォーマットに従って記録する処理を行う。記録先は、例えば、計算機の内部もしくは外部の半導体メモリ、磁気テープ、磁気ディスク、光ディスクなどの物体領域記憶部 10 である。

【0077】

図 6 に、本実施形態における 1 つの物体領域に対する物体領域データの記述フォーマットの一例を示す。

【 0 0 7 8 】

図 6 において、

5 0 1 は、物体 I D で、物体に付与された物体固有の識別情報（例えば、番号または記号）である。

5 0 2 は、参照物体領域一定区間数で、同一の参照物体領域を有するフレーム区間の数である（図 6 では N としている）。これは参照フレームの数とも等しい。

【 0 0 7 9 】

5 0 3 および 5 0 4 は、それぞれ、物体領域存在区間の先頭時刻および末尾時刻である。これらは時刻もしくはフレーム番号により記述される。末尾時刻の代わりに物体領域存在区間の長さ（時刻もしくはフレーム番号の差分値）を用いることもできる。

5 0 5 は、物体領域記述情報である。物体領域記述情報は、参照物体領域同一区間ごとに参照物体領域同一区間数（図 6 の例では N 個）だけ記述される。

【 0 0 8 0 】

各々の物体領域記述情報の具体的内容は、図 6 中の 5 0 6 ～ 5 1 0 に示される。

5 0 6 および 5 0 7 は、それぞれ、参照物体領域同一区間の先頭時刻および末尾時刻である。これらは時刻もしくはフレーム番号により記述される。末尾時刻は参照物体領域同一区間の区間長で置き換えることができる。

5 0 8 は、変換モデル I D である。これは、拡大縮小モデル、a f f i n e 変換モデル、p a r a b o l i c 変換モデルなどのうちどのモデルを用いて物体領域を記述したかを特定するための I D である。

5 1 1 は、原点座標で、変換モデルの原点座標を画像のどこにとるかを決定するものである。原点座標データは、参照物体領域の重心位置を常に原点とするなどのルールを決めておく場合には省略することができる。

5 0 9 は、参照物体領域情報であり、参照物体領域を特定するための情報である。具体的には、参照フレームの時刻（またはフレーム番号）、参照物体領域を表すビットマップ・データ（もしくはビットマップ・データへのポインタ）であ

る。なお、ビットマップ・データは、そのままではデータサイズが大きいため、圧縮して保存するのが好ましい。

510は、変換パラメータ情報である。変換パラメータ情報は、変換モデル（変換モデルID）により定まるパラメータ数（図6の例ではM個）だけ記述される。変換パラメータは、具体的には、各フレームでのパラメータの値を示す配列や、パラメータを近似する関数を特定する情報（係数値など）などである。変換パラメータ情報の詳細については後で詳細に説明する。

【0081】

以上の処理により、映像中の物体の時空間的な領域を簡単な記述データとして記録媒体に記録しておくことができる。

【0082】

なお、本実施形態では、参照フレームを参照物体領域が一定の区間の先頭フレームとしたが、参照フレーム（および参照物体領域）は任意に選択することができる。他の参照物体領域選択方法については、後で（第5の実施形態にて）詳細に説明する。

【0083】

（第2の実施形態）

第2の実施形態は基本的には第1の実施形態と同様であり、以下では第1の実施形態と相違する点を中心に説明する。

【0084】

本実施形態の物体領域データ変換装置の構成例は第1の実施形態（図1）と同様である。

【0085】

図7に、本実施形態の物体領域データ変換装置の処理手順の一例を示す。

【0086】

ステップS201は、図2におけるステップS101と同様の処理である。

【0087】

ステップS202は、物体領域存在区間全体にわたり物体領域を計算する処理である。この具体的な手法としては、Snakesを利用することができる。こ

のように、物体領域存在区間全域にわたって物体領域をあらかじめ求めておく点が第 1 の実施形態と相違するところである。

【 0 0 8 8 】

ステップ S 2 0 3 は、参照物体領域から処理対象となっているフレームの物体領域（対象物体領域）への変換パラメータを算出する処理である。この処理は、図 2 におけるステップ S 1 0 2 のように、オプティカルフローを求める処理と、オプティカルフローから変換パラメータを推定する処理とにより実現できる。これら二つの処理も第 1 の実施形態と同様の処理を用いることができる。

【 0 0 8 9 】

なお、本実施形態においては、変換パラメータを求める前にステップ S 2 0 2 において対象物体領域の形状が算出されているので、物体領域の形状を用いて変換パラメータを算出することも可能である。

具体的な方法としては、まず、対象物体領域を 2 値のビットマップにより表現する。

次に、参照物体領域のビットマップの輪郭部分に隙間無くブロックを配置し、各部ロックごとにテンプレートマッチングにより対象物体領域のビットマップへの移動ベクトルを算出する。

この移動ベクトルをオプティカルフローの代わりとして用いることにより、変換パラメータを算出する。

【 0 0 9 0 】

ステップ S 2 0 4 ～ S 2 0 8 の処理は第 1 の実施形態で説明されている処理と同様である。

【 0 0 9 1 】

本実施形態における物体領域データの記述フォーマットの一例は第 1 の実施形態（図 6）と同様である。

【 0 0 9 2 】

（第 3 の実施形態）

第 1 および第 2 の実施形態では、ビットマップで物体領域を表し、参照物体領域から処理対象となっているフレームの物体領域（対象物体領域）への変換パラ

メータを算出したが、第 3 の実施形態では、物体領域を近似図形で近似し、参照物体領域の近似図形の各代表点から処理対象となっているフレームの物体領域（対象物体領域）の近似図形の対応する各代表点への変換パラメータを算出するものである。本実施形態はこの相違点に対応する部分以外は基本的には第 1、第 2 の実施形態と同様であり、以下では第 1、第 2 の実施形態と相違する点を中心に説明する。

【 0 0 9 3 】

図 8 に、本実施形態に係る物体領域データ変換装置の構成例を示す。図 1 に示されるように、本物体領域データ変換装置は、映像データ記憶部 2、物体領域処理部 4、図形近似処理部 5、変換パラメータ処理部 6、関数近似処理部 8、物体領域データ記憶部 10 を備えている。なお、本処理においてユーザの操作を介入させる形態を取る場合には、映像（動画像）データを例えばフレーム単位で表示させ、ユーザの指示入力等を受け付ける GUI が用いられる（図 8 では省略している）。

【 0 0 9 4 】

図形近似処理部 5 は、前述のように、物体領域を近似図形で近似し、その近似図形の各代表点を求める処理を行うものである。

【 0 0 9 5 】

変換パラメータ処理部 6 は、基準となるフレームの参照物体領域の近似図形の代表点をもとにして、対象となるフレームの物体領域の近似図形の代表点の変換パラメータを算出する処理を行うものとなる。

【 0 0 9 6 】

関数近似処理部 8 は、物体領域の近似図形の代表点の変換パラメータの各々について、その時系列的な軌跡を時間関数に変換する処理を行うことになる。なお、第 1、第 2 の実施形態と同様に、変換パラメータ自体を記述する場合には、この関数近似処理部 8 は不要となる。

【 0 0 9 7 】

もちろん、本物体領域データ変換装置も、計算機上でソフトウェアを実行する形で実現することもできる。

【0098】

図9に、本実施形態の物体領域データ変換装置の処理手順の一例を示す。

【0099】

ステップS301は、図2におけるステップS101と同様の処理である。

【0100】

ステップS302は、物体領域存在区間全体にわたり、物体領域をあらかじめ決められた図形により近似する処理である。この処理は、物体領域を計算する処理と、物体領域を図形近似する処理とにより実現することができる。

【0101】

物体領域を計算する処理は、図7のステップS202と同様の処理により行うことができる。

【0102】

物体領域を図形近似する処理では、例えば、物体領域を囲むなるべく小さな図形を算出する。近似に用いる図形としては、傾きのないあるいは傾きのある矩形（正方形、長方形）、平行四辺形、傾きのないあるいは傾きのある楕円（円を含む）、多角形など、種々の図形を用いることができる。また、領域の近似方法も、領域に外接する図形に近似する方法、領域に内接する図形とする方法、領域の重心を近似図形の重心とする方法、領域と近似図形の面積比を等しくする方法、領域と近似図形とが重ならない部分の面積を最小とする方法など、種々の方法がある。

【0103】

なお、物体の領域を予め定められた図形により近似するのではなく、対象物体（オブジェクト）毎に図形の種類をユーザが指定できるようにしてもよいし、対象物体毎にその物体の形状等に応じて図形の種類を自動的に選択するようにしてもよい。

【0104】

また、物体領域を近似する図形近似を求める処理は、フレームごとに行ってもよいし、もしくは前後数フレームの物体領域を使って図形近似を行ってもよい。後者の場合には、近似図形の大きさや位置などの変化を数フレームの間で平滑化

することにより、近似図形の動きや変形をなめらかにしたり、物体領域の抽出誤差を目立たなくすることができる。なお、近似図形の大きさは、フレームごとに異なって構わない。

【0105】

物体領域の近似図形が求められたならば、この近似図形を表現する代表点を抽出する処理が行われる。どのような点を代表点とするかは、どのような近似図形を用いるかにより異なる。例えば、近似図形が矩形の場合には4つもしくは3つの頂点を代表点とすることができ、近似図形が円の場合には中心と円周上の一点としたり直径の両端点としたりすることができる。また、楕円の場合には楕円の外接矩形の頂点としたり2つの焦点と楕円上の1点（例えば短軸上の1点）としたりすればよい。任意の閉多角形を近似図形とする場合には、各頂点を図形の代表点とする必要がある。

【0106】

代表点の抽出は、1フレーム分の近似図形が得られるたびに、フレーム単位で行う。また、各代表点は、水平方向の座標軸Xと、垂直方向の座標軸Yと、により表される。

【0107】

ここで、図10に、物体の領域が平行四辺形で表される場合に、近似楕円を求める方法の一例を示す。図10における点A, B, C, Dが物体の領域である平行四辺形の各頂点である。この場合、まず、辺ABと辺BCのどちらが長いかを計算する。そして、長い方の辺およびその対辺を辺の一部とする最小の長方形を求める。図10の例の場合は点A, B', C, D'を4頂点とする長方形となる。近似楕円は、例えば、この長方形に内接する楕円と相似で、かつ、点A, B', C, D'を通る外接楕円とする。

【0108】

次に、楕円を表現する図形代表点を2つの焦点および1つの楕円上の点とした場合、楕円の焦点は2つの軸上の点や楕円の外接矩形から簡単に求めることができる。ここでは、一例として、図11における長軸上の2点P0, P1および短軸上の1点Hから焦点FとGを求める方法を説明する。

【0109】

まず、長軸と短軸のパラメータである a 、 b と、楕円の中心 C と、扁平率 e とを、

$$E(P_0, P_1) = 2 \times a、$$

$$C = (P_0 + P_1) / 2、$$

$$E(C, H) = b、$$

$$e = (1/a) \times \sqrt{a \times a - b \times b}$$

により求める。

ここで、 $E(P, Q)$ は点 P と点 Q のユークリッド距離である。

【0110】

このようにして求めたパラメータから、焦点 F と G は、

$$F = C + e \times (P_0 - C)、$$

$$G = C - e \times (P_0 - C)$$

により求めることができる。

【0111】

このようにして、楕円の代表点 F 、 G および H は決定されるが、これらの点を別のフレームにおいて取り出された楕円の代表点と結びつける際にはあいまいさが生じる。すなわち、抽出された2つの焦点と1つ前のフレームにおける2つの焦点とを結びつける組み合わせは2通り存在する。また、短軸と楕円との交点は2つあるため、1つ前のフレームにおいて抽出された楕円上の一点と対応する交点がどちらなのかがわからない。そこで、これらを決定する方法について説明する。

【0112】

1フレーム前に抽出された2つの焦点を F_p 、 G_p とする。 F_p に対応するのが F であるのか G であるのかを判断するために、

$$E((G_p - F_p) / 2, (G - F) / 2) \text{ と}$$

$$E((G_p - F_p) / 2, (F - G) / 2) \text{ とを比較する。}$$

前者の方が小さい場合には、 F_p は F に対応させ、 G_p は G に対応させる。一方、後者の方が小さい場合には、その逆に、 F_p は G に対応させ、 G_p は F に対

応させる。

【0 1 1 3】

また、1つ前のフレームにおける短軸と楕円との交点を H_p とし、現フレームの短軸と楕円との2つの交点を H 、 H' とする。 H_p と対応付ける点として H と H' のどちらを選択するかは2つの距離、

$$E(H_p - (G_p + F_p) / 2, H - (F + G) / 2) \text{ と}$$

$E(H_p - (G_p + F_p) / 2, H' - (F + G) / 2)$ とを算出することにより決定する。

前者が小さい場合には H を選択し、そうでない場合には H' を選択する。なお、はじめのフレームにおける短軸と楕円との交点 H は2つのうちのどちらを選択してもよい。

【0 1 1 4】

次に、近似図形として楕円以外の図形を用いた場合について説明する。

【0 1 1 5】

図12は、近似図形として平行四辺形を用いた場合の代表点を説明するための図である。点 A 、 B 、 C 、 D が平行四辺形の頂点である。これらのうち3点が決まれば残りの1点も一意に決まるため、代表点としては4頂点のうちの3頂点とすれば十分である。この例では、 A 、 B 、 C の3点を代表点としている。

【0 1 1 6】

また、図13は、近似図形として多角形を用いた場合の代表点を説明するための図である。多角形の場合には、頂点の順序を外周に沿った順にしておく。図13の例では、10個の頂点を持つ多角形なので、 $N1 \sim N10$ までの全ての頂点を代表点とする。なお、この場合に、内角が180度未満の頂点のみを代表点とするなどして、頂点数を削減するようにしてもよい。

【0 1 1 7】

さて、上記のように、近似図形は、図形を特定する代表点の集合により表現できる。図14に、幾つかの種類の近似図形の代表点の例をまとめて示す。(a) 矩形、(b) 楕円、(c) 平行四辺形、(d) 多角形の各々について、黒丸で表されているのが代表点である。

【0 1 1 8】

ここで、近似図形の代表点は、位置関係が明らかになるように順序づけられているのが好ましい。図 1 4 の例においては、順序を番号で示しており、矩形および平行四辺形の代表点は、4 頂点のうちの 3 頂点（時計回りに順序づけられる）としている。また、楕円は外接矩形の代表点（時計回りに順序づける）、多角形は全ての頂点（時計回りに順序づける）を図形の代表点とする。

【0 1 1 9】

また、図形を用いて内部に穴の空いた領域を表現することも可能である。図 1 5 に、その表現方法の一例を示す。図 1 5 では、二つの多角形により穴の空いた領域を表現している。図形の代表点は外側および内側の多角形の頂点である。このとき、外側の多角形の頂点は時計回りに順序づけておき、内側の多角形の頂点は反時計回りに順序づけておく。このようにすることにより、常に代表点の進行方向の右側を物体の領域とすることができ、多角形の内部が物体なのか、外部が物体なのかの判定が容易に行えるようになる。

【0 1 2 0】

なお、外側の多角形の頂点を反時計回り、内側の多角形の頂点を時計回りに順序づけてももちろん構わない。この場合には、代表点の進行方向の左側が物体の領域となる。

【0 1 2 1】

続いて、このステップ S 3 0 2 では、参照物体領域と参照フレームを設定する。本実施形態においては、参照物体領域は、物体領域存在区間の先頭のフレームにおける物体領域近似図形である。また、参照物体領域近似図形の代表点の位置を併せて記憶しておく。

【0 1 2 2】

次に、ステップ S 3 0 3 では、処理対象としているフレームの物体領域近似図形の代表点と、参照物体領域近似図形の代表点との対応関係を算出する。

【0 1 2 3】

図 1 6 に、代表点の対応づけ方法の一例を示す。図 1 6 において、1 0 0 0 は近似矩形の重心である。図 1 6 では、参照物体領域近似図形 1 0 0 1 と対象物体

領域近似図形 1 0 0 2 が得られている。

まず、近似図形 1 0 0 1 と近似図形 1 0 0 2 のいずれかを平行移動させ、重心位置を一致させる（重心位置を一致させた状態が図 1 6 である）。

次に、それぞれの矩形の頂点同士の距離 $d_1 \sim d_4$ を計算し、全ての頂点の組み合わせで距離の和を求める。

そして、距離の和が最小となる組み合わせを求め、対応づけする。

なお、この方法では対応付けが難しい場合がある。例えば、近似矩形が正方形に近い形状であり且つ 4 5 度回転移動している場合には、対応付けが難しい（2通りの組み合わせにおいて距離の和が同じような値になる）。

【0 1 2 4】

そこで、そのような場合には、近似図形内の物体領域同士の排他的論理和を取り、その面積が最小となる組み合わせを採用する方法、あるいは物体領域のテクスチャの絶対差分を求め、差分値が最小となる組み合わせを求める方法などを用いれば良い。

【0 1 2 5】

ステップ S 3 0 4 では、物体領域近似図形の代表点の移動ベクトルから変換パラメータを算出する。

【0 1 2 6】

この処理では、代表点の動きをオプティカルフローの代わりに用いて、図 2 のステップ S 1 0 2 と同等の処理により変換パラメータを算出する。ただし、代表点の数は少ないため、必ずしも変換パラメータが求まるとは限らない。例えば、矩形、楕円、平行四辺形は 3 つの代表点を持つが、この 3 つの移動ベクトルから射影変換モデルのパラメータ 8 個を求めることはできない。図 1 7 に、近似に用いる図形の種類と、変換パラメータを求めることができる変換モデルとの関係を示す。図 1 7 中の○はパラメータが算出できる組み合わせ、×は算出できない組み合わせである。

【0 1 2 7】

ステップ S 3 0 5 は、ステップ S 3 0 4 で求めた変換パラメータを時間関数により近似する処理であり、図 2 のステップ S 1 0 3 と同じ処理である。

【 0 1 2 8 】

ステップ S 3 0 6 は、参照物体領域を更新するか否かを判定する処理である。この処理では、まず、変換パラメータにより参照物体領域を変換し、予測物体領域を算出する。もちろん、参照物体領域の代表点のみを変換パラメータにより変換し、変換後の代表点により特定される図形を構成しても同じ予測物体領域が算出できる。次に、予測物体領域と対象物体領域近似図形との誤差を計算し、しきい値処理により参照物体領域更新を行うか否かを判定する。

【 0 1 2 9 】

ステップ S 3 0 7 は、ステップ S 3 0 6 において参照物体領域の更新が必要と判定された場合に、実際に更新を行う処理である。処理対象のフレームの物体領域近似図形を新たな参照物体領域として記憶し、参照物体領域の代表点の座標値も併せて保持しておく。

【 0 1 3 0 】

ステップ S 3 0 8 は、図 2 のステップ S 1 0 6 と同じ処理である。

【 0 1 3 1 】

ステップ S 3 0 9 は、算出した物体領域の情報（物体領域データ）を定められたフォーマットで記録する処理である。

【 0 1 3 2 】

図 1 8 に、物体領域データの記述フォーマットの一例を示す。この記述フォーマットは、図形情報 1 1 0 9 以外は、図 6 に例示した記述フォーマットと同じである。図形情報 1 1 0 9 は、図形の種類を特定する ID と、参照物体領域の図形の代表点の座標とからなる。図 1 8 中の M は ID により特定される図形に必要な代表点の数を表す。

【 0 1 3 3 】

さて、以下では、物体領域の図形近似のバリエーションとして、物体領域の矩形による近似方法について説明する。

【 0 1 3 4 】

例えば、図 1 9 のように物体領域 2 8 0 1 に外接する矩形 2 8 0 2 は、物体領域のビットマップをラスタースキャンし、対象画素が物体領域内であるときに、

X座標とY座標のそれぞれについて、それまで保存されている最小値より小さい場合には最小値を更新し、それまで保存されている最大値より大きい場合には最大値を更新する、という手順を、全ての画素について繰り返しチェックすることによって、X座標とY座標のそれぞれについて物体領域を示す画素位置の最小値および最大値を求めれば、矩形 2 8 0 2 の4つの頂点座標を簡単に得ることができる。

【0 1 3 5】

このような方法は処理が簡易な点で優れているが、例えば図 2 0 のように細長い物体 3 0 0 1 が画面 3 0 0 0 に対して斜めの姿勢で存在しているときには、近似矩形 3 0 0 2 内には非物体領域が特に多く含まれてしまう。また、細長い物体が回転すると矩形 2 8 0 2 の大きさ、形状が変化する。これらは、物体指定する際の弊害となる場合がある。

【0 1 3 6】

そこで、矩形の大きさができるだけ小さくなり（近似矩形内の非物体領域ができるだけ少なくなり）、対象物体の姿勢も反映させることのできる近似方法の一例を示す。

【0 1 3 7】

図 2 1 (a) において、3 1 0 0 は処理対象となっている映像中の 1 フレームを示している。

【0 1 3 8】

3 1 0 1 は抽出対象となっている物体の領域を示している。

【0 1 3 9】

3 1 0 2 は物体の領域を矩形で近似したものである。この近似矩形は図 1 9 の矩形 2 8 0 2 とは異なり、傾きを持っている。矩形内の非物体領域も少なく、対象が回転してもその形状は一定である。

【0 1 4 0】

図 2 2 に、この場合の処理手順の一例を示す。この処理手順例は、対象物体領域の慣性主軸を求め、これに基づいて近似図形を求めるようにしたものである。

【0 1 4 1】

図 2 1 (b) において、3 1 0 3 は、対象物体領域の重心を示している。

【0 1 4 2】

3 1 0 4 は、対象物体領域の慣性主軸を示している。3 1 0 5 は、3 1 0 4 に垂直な直線である。

【0 1 4 3】

まず、対象物体領域の慣性モーメント m_{20} 、 m_{02} 、 m_{11} を求める (ステップ S 7 0 ~ S 7 2)。

【0 1 4 4】

マスク画像を $f(x, y)$ とすると、 $f(x, y)$ は領域内では 1 で、領域外では 0 である。対象領域の慣性モーメントは、

$$m_{ij} = \sum \sum x^i y^j f(x, y)$$

で表せる。

【0 1 4 5】

ここで、原点を通る直線 $y = x \tan \theta$ についての $f(x, y)$ の慣性モーメントは、

$$m_{\theta} = \iint (x \sin \theta - y \cos \theta)^2 f(x, y) dx dy$$

で得られる。

θ を変化させたときに m_{θ} を最小にする角度を θ_0 とする。一通りの角度しかないとき、直線 $y = x \tan \theta_0$ を慣性主軸と呼ぶ。

$\tan \theta_0$ は、2 次方程式、

$$\tan^2 \theta + \{ (m_{20} - m_{02}) / m_{11} \} \tan \theta - 1 = 0$$

の解として求まる。

これにより重心 3 1 0 3 の周りで $\tan \theta_0$ を求めると、対象物体の慣性主軸が得られる (ステップ S 7 3)。

【0 1 4 6】

次に、慣性主軸に平行で物体領域に外接する直線と、慣性主軸に垂直で物体領域に外接する直線を求める (ステップ S 7 4)。

図 2 1 (b) において、直線 3 1 0 6、3 1 0 7 は、慣性主軸 3 1 0 4 に並行な直線であり、対象物体領域に外接する。

直線 3 1 0 8, 3 1 0 9 は、直線 3 1 0 5 に平行な直線であり、対象物体領域に外接する。

矩形 3 1 0 2 は、直線 3 1 0 6, 3 1 0 7, 3 1 0 8, 3 1 0 9 によって形成される (ステップ S 7 5)。

【0 1 4 7】

なお、対象物体が円の場合には慣性主軸が求まらないが、このような場合には例えば前述した物体領域のビットマップをラスタースキャンし、物体領域を示す画素位置の最小値および最大値を求める手順で近似矩形を求めればよい。

【0 1 4 8】

ところで、矩形より楕円で物体領域を表現する方が適当な場合もある。図 2 3 に、物体の領域が矩形で表される場合に、その矩形から近似楕円を求める方法の一例を示す。図 2 4 に、この場合の処理手順の一例を示す。

【0 1 4 9】

図 2 3 において、対象物体領域 3 3 0 0 と外接矩形 3 3 0 1 が得られているものとする。

【0 1 5 0】

まず、近似矩形の内接楕円および外接楕円を求める (ステップ S 8 0)。

【0 1 5 1】

図 2 3 において、楕円 3 3 0 2 は矩形 3 3 0 1 の内接楕円であり、楕円 3 3 0 3 は矩形 3 3 0 1 の外接楕円である。

【0 1 5 2】

次に、内接楕円 3 3 0 2 の大きさを少しずつ外接楕円 3 3 0 3 に近づけていき (ステップ S 8 1)、物体領域を全て包含する楕円 3 3 0 4 を求め (ステップ S 8 2)、近似楕円とする。なお、繰り返し処理において、一回に内接楕円 3 3 0 2 の大きさを拡大する単位は、予め定めておいてもよいし、内接楕円 3 3 0 2 の大きさと外接楕円 3 3 0 3 の大きさの差分に応じて決定してもよい。

【0 1 5 3】

また、上記とは逆に、外接楕円 3 3 0 3 の大きさを内接楕円 3 3 0 2 に近づけていってもよい。この場合には、外接楕円 3 3 0 3 は最初から物体領域を全て包

含しているので、例えば、繰り返し処理において、始めて物体領域に包含されない部分を生じた楕円の、一回前における楕円を、近似楕円とすればよい。

【0 1 5 4】

なお、上記では、物体領域を全て包含する楕円を求めたが、その代わりに、例えば、物体領域の領域と近似楕円の領域とが重複しない領域部分の面積を最小とすることを基準として、近似図形を求めるようにしてもよい。

【0 1 5 5】

次に、近似矩形や近似楕円の代表点を求める。矩形の代表点は4つもしくは3つの頂点を代表点とすることができ、楕円の場合は楕円の外接矩形の頂点としたり、2つの焦点と楕円上の一点としたりすることができる。

【0 1 5 6】

(第4の実施形態)

以下では、第1～第3の実施形態におけるパラメータ情報(図6の510、図18の1110)の具体的な内容について詳細に説明する。

【0 1 5 7】

このパラメータ情報には、参照物体領域に対する各フレームの物体領域の変換パラメータが格納されている。前述したように、変換パラメータを時間関数で表現し、その関数を特定する情報を記述する方法や、変換パラメータの値を直接記述する方法などがあるので、これら表現方法それぞれについて変換パラメータの記述フォーマットを説明する。

【0 1 5 8】

まず、図25に、変換パラメータ情報のデータ構造の一例を示す。

【0 1 5 9】

フレーム数1200は、このパラメータ情報に格納されている物体領域のフレーム数を示す。

【0 1 6 0】

参照物体領域一定区間の各フレームの物体領域の変換パラメータは配列として第1フレームから順に格納される(1201)。

【0 1 6 1】

このような変換パラメータ情報は、1 フレームにつき個数Mだけ存在する。例えば、a f f i n e 変換の場合は、 $M=6$ となる。

【0 1 6 2】

次に、図 2 6 に、変換パラメータを関数を用いて近似した場合の変換パラメータ情報のデータ構造の一例を示す。

【0 1 6 3】

変換パラメータの近似にはスプライン関数を用いるものとする。

【0 1 6 4】

節点フレーム番号 1 3 0 0 は、スプライン関数の節点を表しており、この節点まで多項式のデータ 1 3 0 3 が有効であることを示している。多項式の係数データの数は、スプライン関数の最高次数により変化する（最高次数を K とすると、係数データの数は $K+1$ となる）。そのため、多項式次数 1 3 0 1 を参照する。多項式次数 1 3 0 1 の後には、多項式次数 + 1 個に相当する数の多項式次数 1 3 0 2 が続く。

【0 1 6 5】

また、スプライン関数は節点間で別の多項式で表現されるため、節点の数に対応した数の多項式が必要になる。従って、節点フレーム番号、多項式の係数などを含むデータ 1 3 0 3 は、複数繰り返して記述される。節点フレーム番号が参照物体領域一定区間の最終フレームと等しくなった場合には、それが最後の多項式係数データであることを意味しているので、変換パラメータデータが終わることが分かる。

【0 1 6 6】

次に、図 2 7 に、変換パラメータを関数を用いて近似した場合の物体領域記述情報のデータ構造の他の例を示す。ここでは、多項式の最高次数を 2 次として説明する。

【0 1 6 7】

前述した例（図 6、図 2 6）では、多項式スプライン関数の全ての係数を記述していたのに対して、ここでの記述方法では、スプライン関数の節点の座標と、スプライン関数の 2 次の係数に関連する値との組合せにより、記述する。この記

述方法の利点は、節点が容易に取り出せるため、大まかな物体の軌跡が簡単にわかるという点である。

【0 1 6 8】

以下、この記述方法について詳細に説明する。

【0 1 6 9】

図 2 7 中、変換モデル ID (1 4 0 0) は、物体の動きの近似に用いた変換モデルの種類を特定する。例えば、a f f i n e 変換や射影変換を指定できる。また、原点座標 1 4 1 2 は、変換モデルに従った変換を行う際に、画像のどこを原点とするかを指定するものである。

【0 1 7 0】

参照物体領域情報 1 4 0 1 は、前述の例 (図 6) と同様に参照物体領域の形状を記述する情報が格納される。

【0 1 7 1】

節点数 1 4 0 2 は、スプライン関数の節点の数を表す。各節点に対応するフレームは時間として表され、節点時刻 1 4 0 3 に格納される。節点時刻は、節点数だけあるため、配列として記述しておく (1 4 0 4)。

【0 1 7 2】

同様に、各節点の変換パラメータ (1 4 0 5) もそれぞれパラメータが配列 (1 4 0 6) として記述される。この配列はパラメータの個数 M (a f f i n e 変換の場合は M = 6) だけ存在する。

【0 1 7 3】

一次関数フラグ 1 4 0 7 は、節点間のスプライン関数として一次関数だけがいわれているかどうかを表す。一部分でも 2 次以上の多項式を用いる場合には、このフラグはオフにしておく。このフラグを用いることにより、近似関数として一次関数のみに使われる場合に以下で説明する関数特定情報 1 4 0 8 を一つも記述しなくて済むため、データ量を削減できるというメリットがある。なお、必ずしもこのフラグは必要ではない。

【0 1 7 4】

関数特定情報に含まれる関数 ID (1 4 0 9) と、関数パラメータ (1 4 1 0

) はそれぞれ多項式スプライン関数の次数と、その係数を特定するための情報を表す。図 2 8 に、それらの一例を示す。ここで、 t_a 、 t_b は連続する節点の時刻、 $f(t)$ は $[t_a, t_b]$ の区間のスプライン関数、 f_a 、 f_b は時刻 t_a 、 t_b における節点の変換パラメータを表している。1 次多項式を用いるときは節点のみの情報で十分なので、関数パラメータは記述されないが、2 次多項式の場合には係数を特定するための値一つが関数パラメータに記述される。なお、図 2 8 の例では、2 次の係数が用いられているが、例えば、二次曲線上の f_a 、 f_b 以外の 1 点など、他の値を用いることもできる。

【0 1 7 5】

本実施形態の記述方法では、節点の情報と関数パラメータの情報により、図 2 8 の制約条件を用いて全ての区間におけるスプライン関数が再現できる。

【0 1 7 6】

関数特定情報は、(節点数 - 1) 個存在し、これらは配列となって記述される (1 4 1 1)。

【0 1 7 7】

なお、上記では、多項式の最高次数を 2 次として説明したが、もちろん、多項式の最高次数を 3 次以上とすることも可能である。

【0 1 7 8】

以下では、物体領域データのデータ構造に関連するバリエーションについて説明する。

【0 1 7 9】

まず、以上では、ある 1 つの物体領域に関して全てのフレームを対象として変換パラメータを求めたが、変換パラメータを求めるフレームをサンプリングするようにしてもよい。例えば、3 フレームに 1 フレームだけサンプリングし、フレーム 1 から参照物体領域、フレーム 4、7、… からそれぞれ対象物体領域を用いるなどである。

【0 1 8 0】

なお、変換パラメータを時間関数で表現し、その関数を特定する情報を物体領域データに記述する場合には、そのサンプリングされたパラメータ値によってこ

れまでと同様に関数近似を行えばよい。また、物体領域データにサンプリングに関する情報を含める必要はない。

【0 1 8 1】

一方、変換パラメータの値を物体領域データに直接記述する場合には、例えば、第 1 の実施形態において、(1) そのサンプリングされなかったフレームのパラメータ値を適宜補間し（例えば、直前にサンプリングされたフレームと同じ値を記述する）、物体領域データは図 5 と同様とするか、あるいは、(2) 図 2 9 のように、物体領域データにサンプリング情報 5 2 0 を持たせ、サンプリングした場合にはそのサンプリングしたフレームのパラメータ値のみを記述し、サンプリング情報 5 2 0 にサンプリング方法を特定可能な情報（例えば、 n フレームに 1 回サンプリングしたことを示す情報として数値 n （ただし、例えば $n = 1$ の場合には全てのフレームからサンプリングされたことを示すものとする））を記述すればよい。なお、(2) の方法においては、この物体領域情報を使用する際において、サンプリング情報 5 2 0 を参照することにより、サンプリングされなかったフレームのパラメータ値を必要に応じて補間することができる。以上の点は、他の実施形態の場合も同様である。

【0 1 8 2】

次に、以上の構成において、参照物体領域の更新判定処理を行わずに、参照物体領域を固定する構成も可能である。例えば、物体領域の動きが少ない場合や、あるいは、ある程度誤差が大きくても良いような場合などには、処理が簡易になるの有効である。この場合、物体領域データのデータ構造はこれまでのものと同じものを使用することができる（参照物体領域一定区間数が 1 となり、物体領域記述情報が 1 個となる）。また、常に参照物体領域の更新判定処理を行わない場合には、物体領域データのデータ構造はこれまでのものから参照物体領域一定区間数のフィールドを削除し、物体領域記述情報のフィールドを 1 個とするようにしてもよい。

【0 1 8 3】

（第 5 の実施形態）

続いて、これまでの各実施形態における参照物体領域選択方法の他の例につい

て説明する。

【0184】

図30は、予測物体領域の誤差が最小となるように、物体領域存在区間から参照物体領域を選択する手法の一例を表したフローチャートである。

【0185】

ステップS400では、物体領域存在区間から1フレーム分の物体領域データを参照物体領域として取り出す。

【0186】

ステップS401では、ステップS400の参照物体領域に対して、物体領域存在区間の他フレームの物体領域の変換パラメータを求める。

【0187】

ステップS402では、ステップS400の参照物体領域とステップS403で求めた変換パラメータを用いて、物体領域存在区間の参照フレームを除く全フレームの予測物体領域を求める。

【0188】

ステップS403では、物体領域存在区間の参照フレームを除く全フレームについて、ステップS402で求めた予測物体領域と実際の物体領域との誤差を求め、それらを加算して誤差の合計値を求める。

【0189】

ステップS404では、ステップS403で求めた誤差の合計値が、以前の他のフレームを参照フレームとして求めた誤差の合計値のいずれよりも小さい場合は、現在の参照フレーム番号を保存する。

【0190】

ステップS405は、分岐処理であり、現在の参照フレームが物体領域存在区間の最終フレームである場合にはステップS406へ、そうでない場合にはステップS400へと分岐する。ステップS400へ分岐した場合には、新たなフレームを参照フレームとして、同様の処理を繰り返す。

【0191】

ステップS406では、保存してあった予測物体領域と実際の物体領域との誤

差の合計値が最小となる参照フレームの番号を取り出して、このフレームの物体領域を参照物体領域とする。

【0192】

図30を用いた実施形態は計算量が大きいため、より高速な処理が求められる場合には、精度を犠牲にして、特定フレームを参照フレームとして選択すればよい。このフレームは、物体領域存在区間の先頭のフレームであってもよいが、中央のフレームを参照フレームとして用いれば、参照フレームと対象フレームの間隔のうち、最も離れた部分が最小となるため、多くの場合にはより精度のよい近似が期待できる。

【0193】

図31は、物体領域存在区間の中央のフレームを参照フレームとして、参照物体領域を選択する手法の一例を説明する図である。

【0194】

物体領域存在区間1700の先頭フレーム1701と最終フレーム1702の中央フレーム1703を参照フレームとして選択して、他のフレームの物体領域の変換パラメータを求めている。

【0195】

図32は、物体領域存在区間から、複数の参照フレームをリアルタイムに選択する手法の一例を表したフローチャートである。また、図33は、同手法を説明する図である。リアルタイム処理の場合、最終的な物体領域存在区間が確定していなくても、参照フレームを決定することが必要である。

【0196】

ステップS500では、物体領域存在区間から1フレーム分の物体領域データを参照物体領域として取り出す。図33中では先頭フレーム1900に相当する。

【0197】

ステップS501は、分岐処理であり、現在の処理フレームが物体領域存在区間の最終フレームである場合には終了へ、そうでない場合にはステップS502へと分岐する。

【0198】

ステップS502では、物体領域存在区間から新たな物体領域データを対象物体領域として取り出し、ステップS501の参照物体領域に対する変換パラメータを求める。

【0199】

ステップS503では、ステップS501の参照物体領域とステップS502で求めた変換パラメータを用いて予測物体領域を求める。

【0200】

ステップS504は、分岐処理であり、ステップS503で求めた予測物体領域とステップS502の実際の対象物体領域との誤差が閾値を越えているかを判定する。誤差が閾値を越えている場合はステップS505へ、越えていない場合はステップS501へ分岐する。

【0201】

ステップS501へ分岐した場合、現在のフレームが物体領域存在区間の最終フレームでなければ同様な処理を繰り返す。これらの同じ参照フレームに対して、繰り返し処理されるフレームの集合は図33中では1901に相当する。

【0202】

ステップS505では、現在のフレームを新たな参照フレームとし、ステップS501へ進んで、現在のフレームが物体領域存在区間の最終フレームでなければ同様な処理を繰り返す。図33中では新たな参照フレーム1902に相当する。

【0203】

参照フレームが更新されると、参照物体領域一定区間1903が確定する。物体領域存在区間は複数の参照物体領域一定区間によって構成される。

【0204】

図34は、物体領域存在区間から、複数の参照フレームを選択する別の手法を表したフローチャートである。また、図35は、同手法を説明する図である。

【0205】

ステップS600では、未処理の物体領域存在区間から中央のフレームの物体

領域データを参照物体領域として取り出す。図 3 5 (a) 中では物体領域存在区
間 2 1 0 0 の中央フレーム 2 1 0 1 が参照フレームとなる。

【 0 2 0 6 】

ステップ S 6 0 1 は、分岐処理であり、現在処理しているフレームが未処理の
物体領域存在区間の先頭フレームかどうかを判定する。先頭フレームでない場合
にはステップ S 6 0 2 へ、先頭フレームである場合にはステップ S 6 0 6 へと分
岐する。

【 0 2 0 7 】

ステップ S 6 0 2 では、現在の処理フレームの 1 つ前のフレームの物体領域を
取り出し、ステップ S 6 0 0 の参照物体領域に対する変換パラメータを求める。

【 0 2 0 8 】

ステップ S 6 0 3 では、ステップ S 6 0 0 の参照物体領域とステップ S 6 0 2 で
求めた変換パラメータを用いて、予測物体領域を求める。

【 0 2 0 9 】

ステップ S 6 0 4 は、分岐処理であり、ステップ S 6 0 3 で求めた予測物体領
域とステップ S 6 0 2 の実際の対象物体領域の誤差が閾値を越えているかを判定
する。誤差が閾値を越えている場合はステップ S 6 0 5 へ、越えていない場合は
ステップ S 6 0 1 へ分岐する。ステップ S 6 0 1 へ分岐した場合、現在のフレー
ムが未処理の物体領域存在区間の先頭フレームでなければ同様な処理を繰り返す
。

【 0 2 1 0 】

ステップ S 6 0 5 では、注目フレームの 1 つ後のフレームを参照物体領域一定
区間の先頭フレームとする。図 3 5 中では、参照フレーム 2 1 0 1 の物体領域を
参照物体領域として処理したときに、フレーム 2 1 0 2 (第 k フレームとする)
の物体領域で誤差が閾値を越えた場合、第 k + 1 フレームが参照物体領域一定区
間の先頭フレームとなる。

【 0 2 1 1 】

ステップ S 6 0 6 では、注目しているフレームをステップ S 6 0 0 で取り出し
たフレームと同じ未処理の物体領域存在区間の中央のフレームに戻る。

【0 2 1 2】

ステップ S 6 0 7 ~ S 6 1 1 は前述のステップ S 6 0 1 ~ 6 0 5 と同様の処理を、後のフレームに向かって行う。図 3 5 中で、フレーム 2 1 0 3 の直前のフレームが参照物体領域一定区間の最終フレームになったとすると、未処理の物体領域存在区間 2 1 0 5 と 2 1 0 6 が残る。

【0 2 1 3】

ステップ S 6 1 2 は、分岐処理であり、未処理の物体領域存在区間が存在していればステップ S 6 0 0 へ、存在していなければ処理を終了する。ステップ S 6 0 0 へ分岐した場合、未処理の物体領域存在区間の中央フレームを新たな参照フレームとして、未処理の物体領域存在区間がなくなるまで、処理を繰り返す。図 3 5 (b) 中では、未処理の物体領域存在区間 2 1 0 5 の中央フレーム 2 1 0 7 を新たな参照フレームとして処理し、新たな参照物体領域一定区間 2 1 0 8 と未処理の物体領域存在区間 2 1 0 9 が生じた例を示している。

【0 2 1 4】

(第 6 の実施形態)

続いて、これまでの各実施形態において、1 つの物体 (オブジェクト) を複数の領域に分けて物体領域データを作成する方法について説明する。

【0 2 1 5】

これまでは、1 つの物体に対して 1 つの変換パラメータを求めていた。しかし、見かけの形状が大きく変化するような物体の場合、物体をいくつかの領域に分割し、それぞれの領域に対して変換パラメータを用いる方が良い場合がある。例えば、歩行している人間では、頭や胴体の部分は動きが少ないが、手足は激しく動作する。人間を 1 つの物体として扱うよりも、頭・胴体・手・足と別の領域に分割する方が、各部の変換パラメータを安定して求めることができる。

【0 2 1 6】

1 つの物体を複数の図形で表わす場合には、物体を複数の領域に分割する処理が必要となる。この処理にはどのような方法を用いてもよいが、例えば人手で直接入力する方法がある。この場合、例えば、マウス等のポインティングデバイスを用いて、画像上で領域を矩形や楕円で囲む、あるいはポインティングデバイス

の軌跡により領域を指定する、などの操作で実現することができる。また、人手ではなく自動で行う場合には、例えば、物体の動きのクラスタリングで実現する方法がある。これは、連続するフレーム間で物体中の各領域がどのような動きをしたかを相関法（例えば、画像解析ハンドブック、第II部、第3章、東京大学出版会、1991を参照）や勾配法（例えば、Determining optical flow, B. K. P. Horn and B. G. S. Chumck, Artificial Intelligence, Vol. 17, pp. 185-203; 1981を参照）などにより求め、これらの動きの似ているものだけをまとめて領域を形成する方法である。

【0217】

図36に、オプティカルフローの似ているものをまとめて領域分割を行う様子を示す。

【0218】

また、図37に、物体を複数の領域で記述するためのデータ構造の一例を示す。これは物体を単一の領域で記述するデータ構造（図18）を拡張したものであり、2906以下のデータは図18と共通である。2902に分割した領域の数を保持し、2905以降に各領域のデータを保持する。さらに、図6に示すように穴の空いた領域についても、穴を1つの領域として表現し、図37のデータ構造で扱うことが可能である。

【0219】

（第7の実施形態）

次に、これまでの各実施形態により得られる物体領域データの利用例について説明する。

【0220】

例えば、物体領域データに該物体（オブジェクト）に関連する関連情報を付加するようにしてもよい。この場合、映像を閲覧中のユーザが映像中の物体を指示する（例えばマウスでクリックする）ことにより、該物体の関連情報を呈示するような、ハイパーメディア・アプリケーションを実現することができる。

【0221】

その際、映像や物体はどのようなものであってもよい。例えば、映像が映画などのコンテンツ、物体が俳優等の登場人物もしくは他の登場物体、関連情報がその俳優もしくはその役柄等に関する説明であれば、映画を視聴している視聴者は、所望の俳優に関する説明をその俳優の画像をクリックするだけで閲覧することができる。同様に、電子百科事典、電子カタログ等のあらゆる電子コンテンツに適用可能である。

【0222】

関連情報は、文字、音声、静止画、動画、あるいはそれらを適宜組み合わせたものであってもよいし、プログラムもしくは計算機の動作を記述したデータであってもよい。前者の場合には情報が呈示され、後者の場合にはプログラム等が実行される。

【0223】

物体領域データには、関連情報を付加する代わりに、関連情報データ自体であってもよいし、その関連情報を取得するためのアドレス等のポインタ情報であってもよい。

【0224】

また、例えば、関連情報にキーワードを記述しておけば、物体をキーワード検索することができる。さらに、関連情報にその物体から抽出した、シェープ、テクスチャ、アクティビティ、カラーなどの特徴量を記述しておけば、そのような特徴量をもとにオブジェクト検索することができる。

【0225】

また、例えば、物体領域データを解析することにより得られる、物体のシェープ、テクスチャ、アクティビティ、カラーなどの特徴量に基づいて、不審な人物等の監視を行う、監視システムを実現することができる。

【0226】

次に、映像データや物体領域データの提供方法について説明する。

【0227】

本実施形態の処理により作成された物体領域データがユーザの用に供される場合には、作成者側からユーザ側に何らかの方法で物体領域データを提供する必要

がある。この提供の方法としても以下に例示するように種々の形態が考えられる。

(1) 映像データとその物体領域データとその関連情報とを 1 つ (または複数の) 記録媒体に記録して同時に提供する形態

(2) 映像データとその物体領域データとを 1 つ (または複数の) 記録媒体に記録して同時に提供するが、関連情報は別途提供するかもしれない (後者は例えば提供しなくてもユーザがネットワーク経由等で別途取得できる場合) 形態

(3) 映像データを単独で提供し、別途、物体領域データと関連情報とを 1 つ (または複数の) 記録媒体に記録して同時に提供する形態

(4) 映像データ、物体領域データ、関連情報を別々に提供する形態

上記は主に記録媒体により提供する場合であるが、その他にも、一部または全部を通信媒体で提供する形態も考えられる。

【0 2 2 8】

(第 8 の実施形態)

以下では、本発明の物体領域表現方法をハイパーメディアに応用した例について説明する。

【0 2 2 9】

物体 (オブジェクト) に関連情報を結び付けるハイパーリンクでは、物体がユーザによって指定されたことを検出する必要がある。ここでは指定された点が時空間物体領域の内部にあるかあるいは外部にあるかを判定する方法の一例を示す。

【0 2 3 0】

なお、本実施形態では、ユーザが (主に G U I 画面上で) 物体を指示することにより、関連情報を呈示 (文字、静止画、動画等の表示、音声の出力等) させ、あるいは関連するプログラムを実行させるような場合を例にとって説明する。

【0 2 3 1】

図 3 8 に、本実施形態に係る情報処理装置の構成例を示す。図 3 8 に示されるように、本情報処理装置は、映像データ表示部 3 0 1、制御部 3 0 2、関連情報

呈示部 3 0 3、指示入力部 3 0 4 を備えている。

【 0 2 3 2 】

映像データ表示部 3 0 1 は、図示しない記録媒体等から入力した映像データを液晶表示装置もしくは C R T 等に表示するためのものである。

【 0 2 3 3 】

指示入力部 3 0 4 は、ユーザがマウス等のポインティングデバイスもしくはキーボードなどを用いて、液晶表示装置もしくは C R T 等に表示された映像中の物体を指示するなどの操作を行うことを可能とし、そのユーザからの入力を受け付けるためのものである。

【 0 2 3 4 】

制御部 3 0 2 は、詳しくは後述するが、例えばユーザが画面上で指示した座標と図示しない記録媒体等から入力した物体領域データとに基づいてユーザが映像中の物体を指示したか否か判定する。

【 0 2 3 5 】

関連情報呈示部 3 0 3 は、制御部 3 0 2 により物体が指示されたと判定されたときに、物体領域データに関連情報が付加されている場合には該関連情報を呈示し、物体領域データに関連情報へのポインタ情報が付加されている場合には該ポインタ情報をもとに関連情報を（記録媒体やネットワークを介したサーバ等から）取得して呈示する。

【 0 2 3 6 】

なお、関連情報がプログラムや計算機の動作を記述したデータである場合には、当該プログラム等が実行される。

【 0 2 3 7 】

また、物体が指示された場合に、関連情報の呈示等を行うのではなく、他のアクションが取られる場合には、関連情報呈示部 3 0 3 の代わりに該当する処理部が実装される。

【 0 2 3 8 】

もちろん、本実施形態もソフトウェアによっても実現可能である。

【 0 2 3 9 】

さて、以下、図 3 9 の例を用いて、本実施形態の処理手順について説明する。

【 0 2 4 0 】

図 3 9 において、参照物体領域を 2 2 0 1、対象フレームを 2 2 0 2 とする。ユーザによって指定された点が 2 2 0 3 であり、その点の参照フレームにおける位置が 2 2 0 4 である。

【 0 2 4 1 】

図 4 0 に、本実施形態の処理手順の一例を示す。ただし、図 4 0 のフローチャートでは、映像の再生中に表示されている映像内をマウスカーソル等のポインティングデバイスを用いて指示された際に指定された点が時空間物体領域の内部にあるかあるいは外部にあるかを判定する処理についてのみ示している（基本的には制御部 3 0 2 の処理に相当するものである）。

【 0 2 4 2 】

ステップ S 7 0 0 として、対象フレーム番号および指定された点の座標を取得する。すなわち、ポインティングデバイス等により指示された画面上の座標が、映像中の画像のどこに相当するかを計算する。さらに、指示された瞬間に再生を行っていた映像のフレーム番号を取得する。なお、フレーム番号ではなく、タイムスタンプを用いてもよい（以下では、フレーム番号として説明する）。

【 0 2 4 3 】

ここで、映像に付随している映像中の物体の物体領域データから、上記対象フレーム番号において映像中に存在している物体を選択するものとする。これは、物体領域データにおける先頭フレーム番号および末尾フレーム番号を参照することにより容易に実行できる。

【 0 2 4 4 】

当該フレームに（指示できる）物体が存在しない場合には、この時点で処理は終了となる。

【 0 2 4 5 】

なお、以下の処理は、対象となる物体が複数存在する場合には、各物体について、同時にまたは例えば指示された物体が得られるか最後の物体になるまで順次に繰り返し行うものとする。

【0 2 4 6】

ステップ S 7 0 1 では、対象物体について、近似変換パラメータ軌跡から変換パラメータを算出する。この変換パラメータは、参照物体領域から対象物体領域への変換を示している。

【0 2 4 7】

ステップ S 7 0 2 では、指定された点を逆変換して、参照フレームにおける位置を求める。

【0 2 4 8】

例えば、変換が *a f f i n e* 変換である場合に、図 3 9 において点 2 2 0 3 から点 2 2 0 4 を求めるには、数式 (8) を用いればよい。

【0 2 4 9】

【数 2】

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad \dots (8)$$

【0 2 5 0】

点 2 2 0 3 の座標を (x', y') 、点 2 2 0 4 の座標を (x, y) とする。

【0 2 5 1】

ステップ S 7 0 3 では、その点が参照物体領域の内部にあるか外部にあるかを判定する。

【0 2 5 2】

内外判定の方法は、使用されている物体領域の表現方法によって異なる。以下では、ビットマップ表現と多角形表現の場合を例として内外判定の方法を説明する。

【0 2 5 3】

参照物体領域がビットマップで表現されているとき、図 4 1 の手順に従って、内外判定を行う。参照物体領域のビットマップ表現とは、例えば、物体領域内を 1、物体領域外を 0 で表現することがあげられる。ステップ S 8 0 2 によって算

出された点の位置のビットマップ情報を、ステップ S 803 で参照して値が 1 のとき物体領域内にあり、0 のとき物体領域外にあると判定できる。

【0254】

また、参照物体領域が多角形で表現されているとき、図 4 2 の手順に従って、内外判定を行う。例えば、図 4 3 に示す参照物体領域多角形において、点 2501 と多角形の 2 つの隣り合った頂点を結ぶベクトル 2502 および 2503 を生成する。ベクトルは 3 次元とし、Z 座標を 0 としておく。また、ベクトルの長さが 1 となるように正規化しておく。2 つのベクトルの外積およびベクトル同士のなす角を、1 つずつ頂点をずらしながら全ての頂点について算出する。外積ベクトルは多角形に対して垂直であり、Z 座標は -1 か 1 の値を取る。算出した角度に外積ベクトルの Z 座標をかけあわせ加算していくと、点が多角形内部にある場合は -360 度もしくは 360 度になり、多角形外部にある場合はそれ以外の値をとる。以上のようにして内外判定を行うことができる。

【0255】

また、参照物体領域が順序のついた代表点で表現されている場合に、それをいったんビットマップに展開してから、内外判定を行う方法も可能である。

【0256】

まず、参照物体領域の特徴点をその順序に従って結び多角形を生成する。参照物体領域が多角形表現されている場合に内外判定を行う方法については、上述の通りである。生成した多角形をビットマップに展開するには、多角形の全ての辺をあらわす直線の式を求める。それらの式から、多角形内部を表現する連立不等式をたてる。

【0257】

簡単な例として多角形が長方形である場合を図 4 4 に示す。長方形の各辺の式が図 4 4 に示すように得られているとき、多角形内部を表現する連立不等式は数式 (9) のように求められる。

【0258】

【数 3】

$$\left\{ \begin{array}{l} x \geq 2 \\ x \leq 7 \\ y \geq 1 \\ y \leq 4 \end{array} \right. \quad \dots (9)$$

【0 2 5 9】

ビットマップの各画素の座標がこの連立不等式を満たせば、その画素の値を 1 とし、そうでない場合は画素値を 0 とすることによって、ビットマップへ展開することが可能である。展開されたビットマップを用いて内外判定を行う方法については上述の通りである。

【0 2 6 0】

さて、以上のような判定の結果、ある物体領域が指示されたことが検出された場合には、それを契機として所定の処理が行われる。

【0 2 6 1】

例えば、物体領域データに含まれている関連情報へのポインタを参照し、このポインタ情報に基づいて関連情報を取得し、その表示等を行う（図 3 8 の構成例では、これを関連情報呈示部 3 0 3 により行う）。また、関連情報としてプログラムが指定されている場合には、指定プログラムを実行したり、またその他、指定されている所定の動作を行う。なお、物体領域データに関連情報そのものが記述されている場合にはこれを表示等すればよい。

【0 2 6 2】

図 4 5 に、関連情報として映像中の物体の説明が付与されている場合の一例を示す。映像 8 0 0 の再生中にポインティングデバイス 8 0 2 により指示された座標が物体 8 0 1（を近似した図形の）領域内部であった場合、関連情報 8 0 3 が表示される。

【0 2 6 3】

なお、以上の各機能は、ソフトウェアとしても実現可能である。

【0 2 6 4】

また、本実施形態は、コンピュータに所定の手段を実行させるための（あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュータに所定の機能を実現させるための）プログラムを記録したコンピュータ読取り可能な記録媒体としても実施することもできる。

【0265】

本発明は、上述した実施の形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。

【0266】

【発明の効果】

本発明によれば、映像中における対象となる物体の領域を、参照物体領域と変換パラメータに基づいて記述することにより、映像中の所望の物体の領域を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にすることができる。

【0267】

また、本発明によれば、ユーザによる映像中の物体の指示の判定を容易にすることができる。

【図面の簡単な説明】

【図1】

本発明の実施の形態に係る物体領域データ変換装置の構成例を示す図

【図2】

同実施形態における処理手順の一例を示すフローチャート

【図3】

物体領域のオプティカルフロー算出方法の一例を説明するための図

【図4】

物体領域のオプティカルフロー算出方法の他の例を説明するための図

【図5】

1つの変換パラメータを時間関数で近似表現する例を説明するための図

【図6】

参照物体領域がビットマップ表現されている場合の物体領域データの記述フォ

ーマットの一例を示す図

【図 7】

同実施形態における処理手順の他の例を示すフローチャート

【図 8】

物体領域データ変換装置の他の構成例を示す図

【図 9】

同実施形態における処理手順のさらに他の例を示すフローチャート

【図 1 0】

物体の領域を楕円で近似する例について説明するための図

【図 1 1】

物体の領域を近似する楕円の代表点を求める例について説明するための図

【図 1 2】

近似図形を平行四辺形とした場合の代表点の例について説明するための図

【図 1 3】

近似図形を多角形としたときの代表点の例について説明するための図

【図 1 4】

物体領域近似図形と代表点の例を示す図

【図 1 5】

穴のある物体領域を近似図形で表した例を示す図

【図 1 6】

物体領域の近似図形の代表点同士を対応付ける方法について説明するための図

【図 1 7】

近似に用いる図形の種類と変換パラメータを求めることができる変換モデルとの関係を示す図

【図 1 8】

参照物体領域が図形近似されている場合の物体領域データの記述フォーマットの一例を示す図

【図 1 9】

物体を矩形で近似した様子を示す図

【図 2 0】

傾斜した細長い物体を傾斜を持たない矩形で近似した様子を示す図

【図 2 1】

物体をその傾斜に応じた傾斜を持つ矩形で近似した様子を示す図

【図 2 2】

近似矩形を求める処理手順の一例を示すフローチャート

【図 2 3】

近似矩形から近似楕円を求める方法について説明するための図

【図 2 4】

近似矩形から近似楕円を求める処理手順の一例を示すフローチャート

【図 2 5】

物体領域データ中のパラメータ情報のデータ構造の一例を示す図

【図 2 6】

近似関数を用いた場合のパラメータ情報のデータ構造の一例を示す図

【図 2 7】

物体領域記述情報のデータ構造の他の例を示す図

【図 2 8】

関数 ID と関数形式と関数パラメータと制約条件との対応の一例を示す図

【図 2 9】

サンプリング情報を含む物体領域データの記述フォーマットの一例を示す図

【図 3 0】

誤差最小となる参照物体領域を選択する処理手順の一例を示すフローチャート

【図 3 1】

物体領域存在区間の中央フレームを参照物体領域とする処理手順の一例を説明するための図

【図 3 2】

参照物体領域一定区間の先頭を参照物体領域とする処理手順の一例を示すフローチャート

【図 3 3】

参照物体領域一定区間の先頭を参照物体領域とする処理手順の一例を説明するための図

【図 3 4】

参照物体領域一定区間の中間を参照物体領域とする処理手順の一例を示すフローチャート

【図 3 5】

参照物体領域一定区間の中間を参照物体領域とする処理手順の一例を説明するための図

【図 3 6】

オプティカルフローを用いて 1 つの物体を動きが似ている領域に分割する様子を説明するための図

【図 3 7】

1 つの物体を複数の領域で記述するための物体領域データの記述フォーマットの一例を示す図

【図 3 8】

同実施の形態に係る情報処理装置の構成例を示す図

【図 3 9】

指定された位置情報を持つ点が時空間物体領域の内部にあるか外部にあるかを判定する方法について説明するための図

【図 4 0】

指定された位置情報を持つ点が時空間物体領域の内部にあるか外部にあるかを判定するための処理手順の一例を示すフローチャート

【図 4 1】

参照物体領域がビットマップ表現されているときの内外判定方法の一例を示すフローチャート

【図 4 2】

参照物体領域がポリゴン表現されているときの内外判定方法の一例を示すフローチャート

【図 4 3】

ポリゴンと点の内外判定を行う方法について説明するための図

【図 4 4】

ポリゴンで表現されている参照物体領域をビットマップ表現に展開する方法について説明するための図

【図 4 5】

物体領域データを利用したハイパーメディア・コンテンツの表示例を示す図

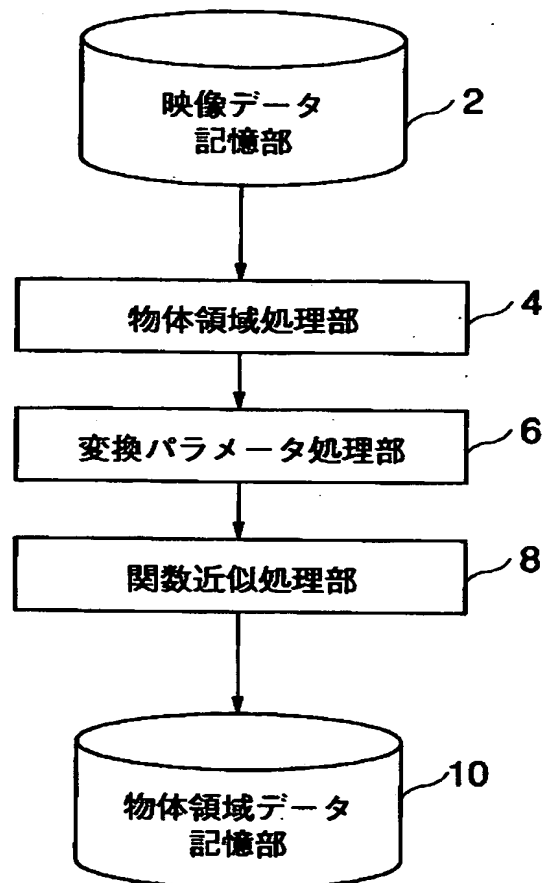
【符号の説明】

- 2 … 映像データ記憶部
- 4 … 物体領域処理部
- 5 … 図形近似処理部
- 6 … 変換パラメータ処理部
- 8 … 関数近似処理部
- 1 0 … 領域データ記憶部
- 3 0 1 … 映像データ表示部
- 3 0 2 … 制御部
- 3 0 3 … 関連情報呈示部
- 3 0 4 … 指示入力部

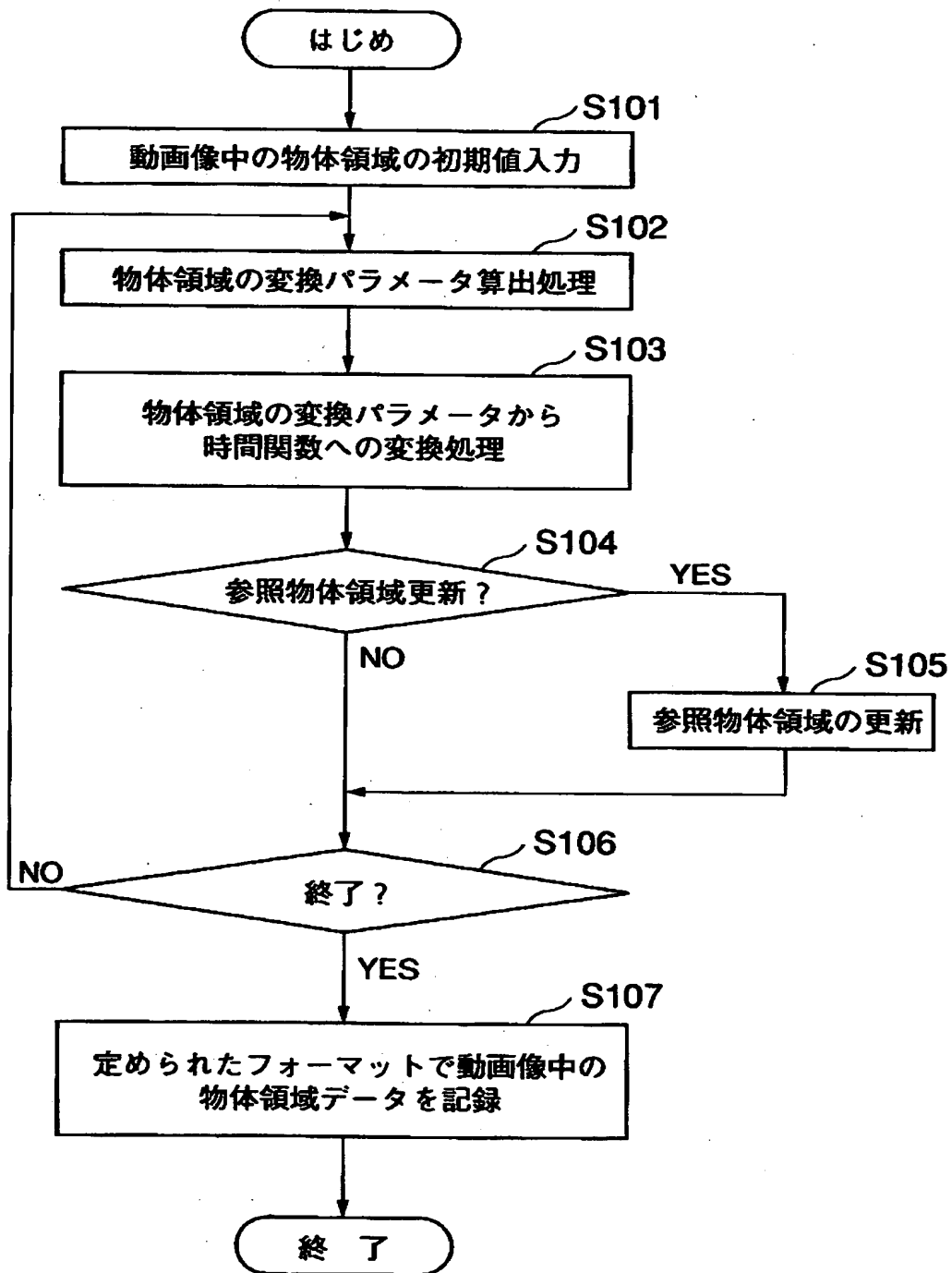
【書類名】

図面

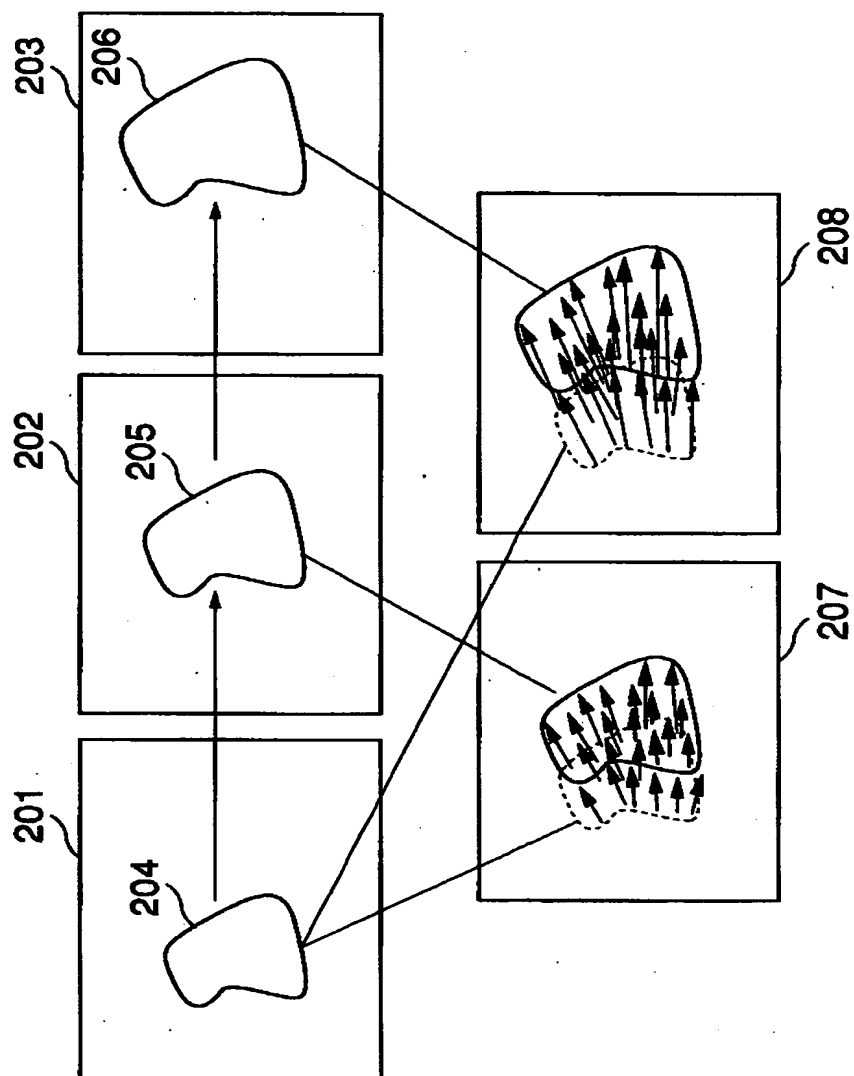
【図 1】



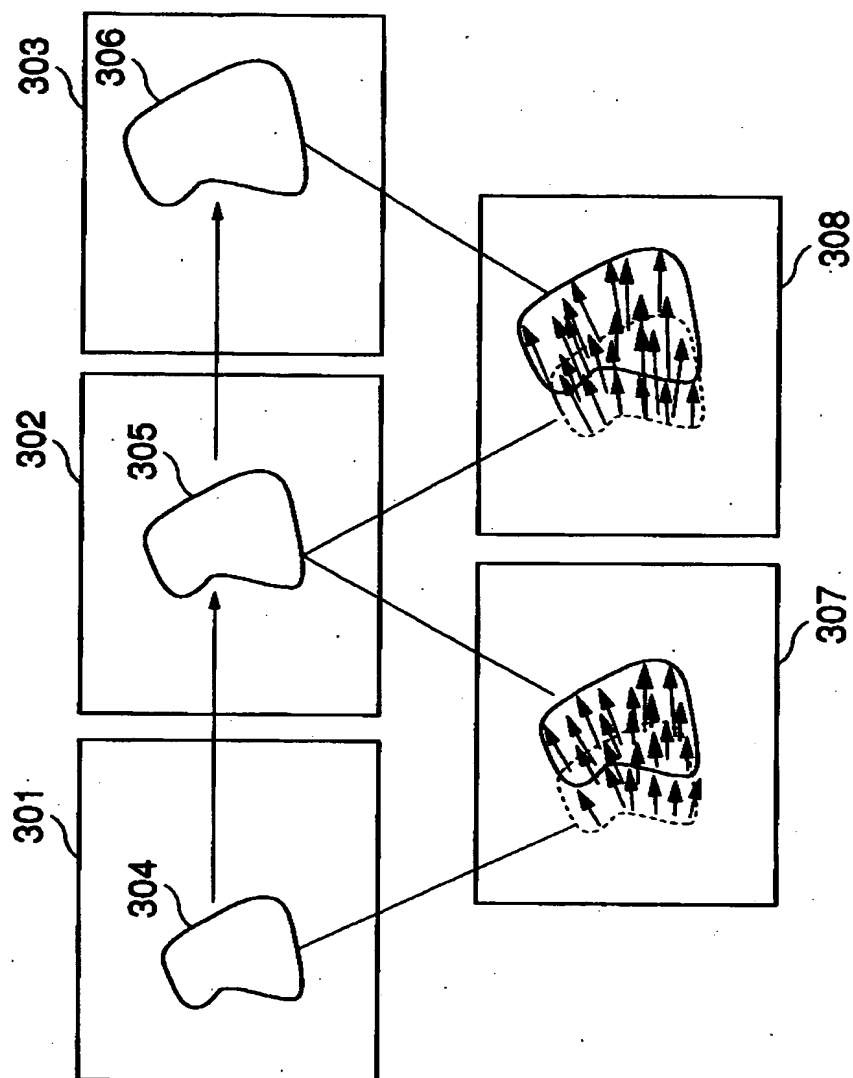
【図 2】



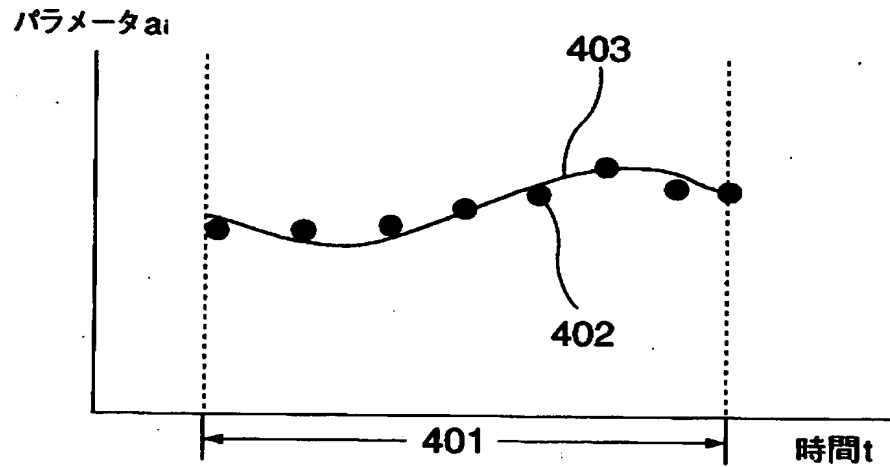
【図 3】



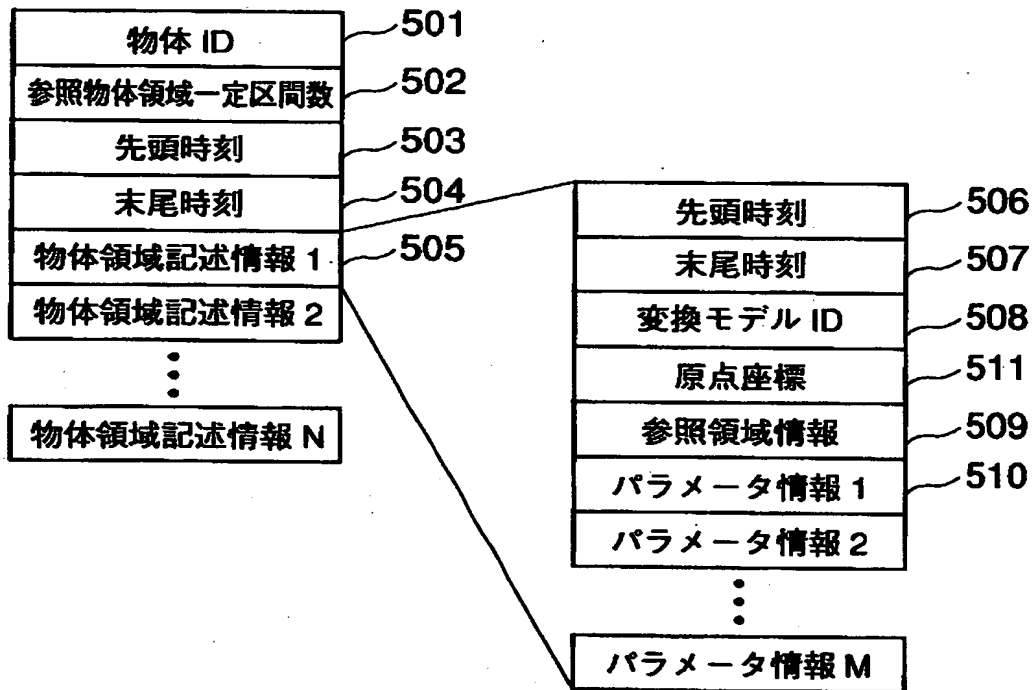
【图 4】



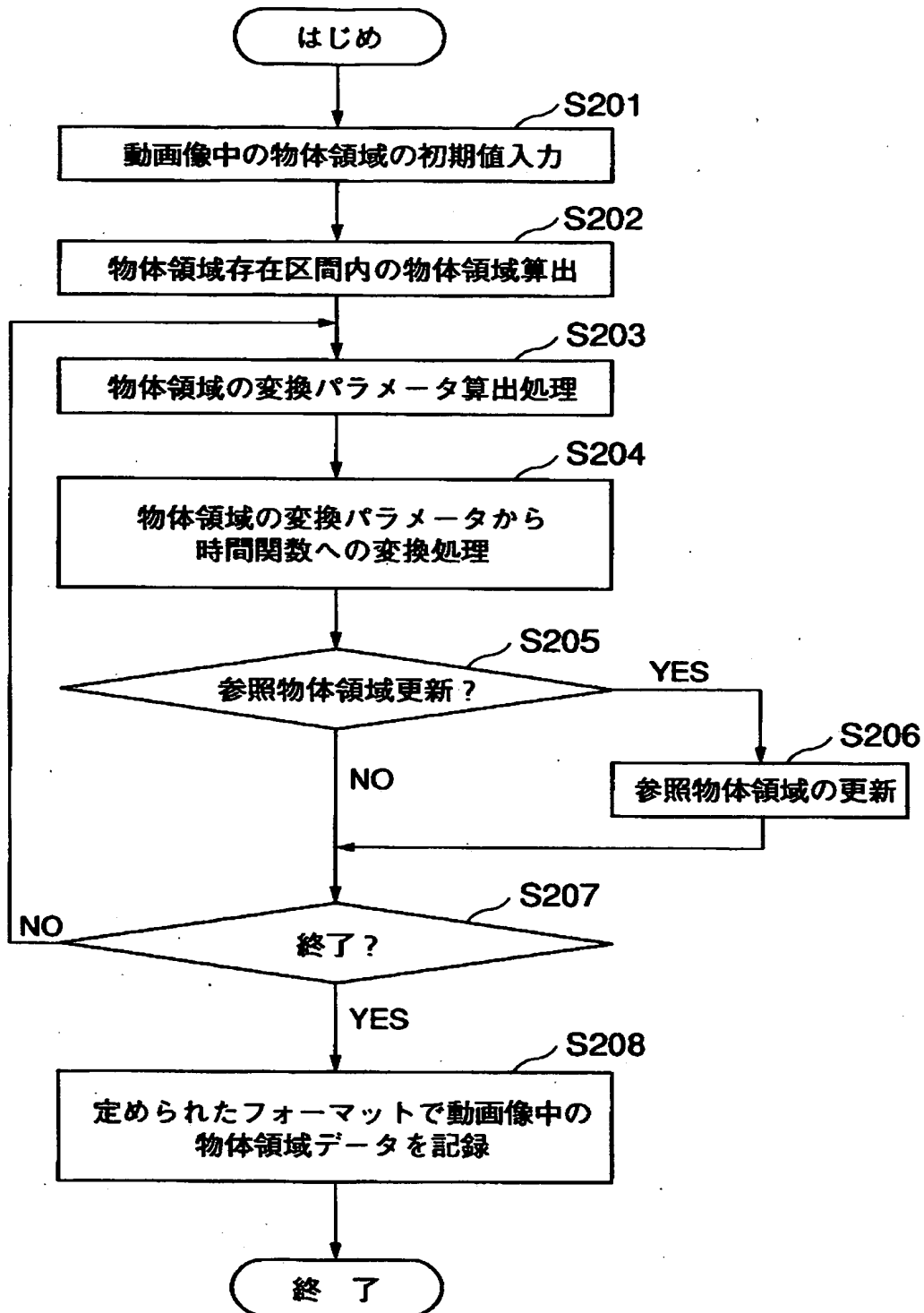
【図 5】



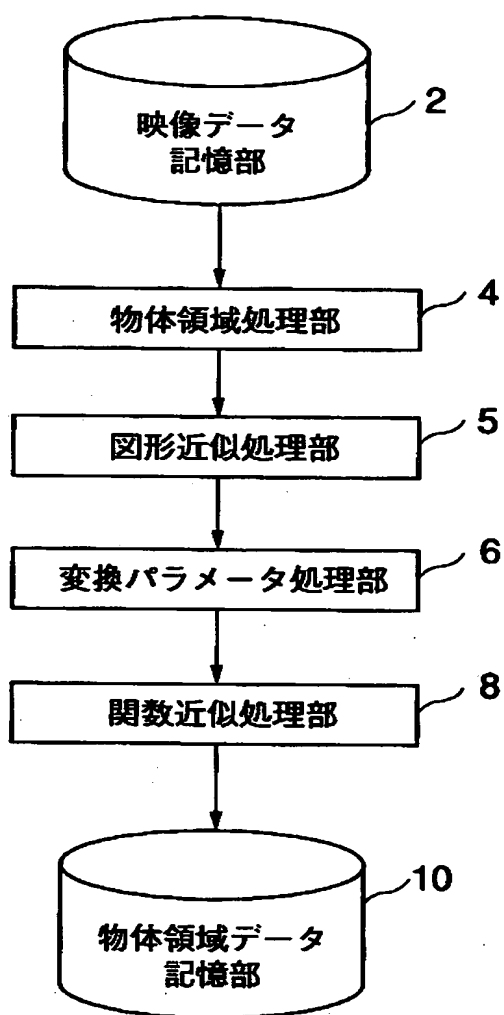
【図 6】



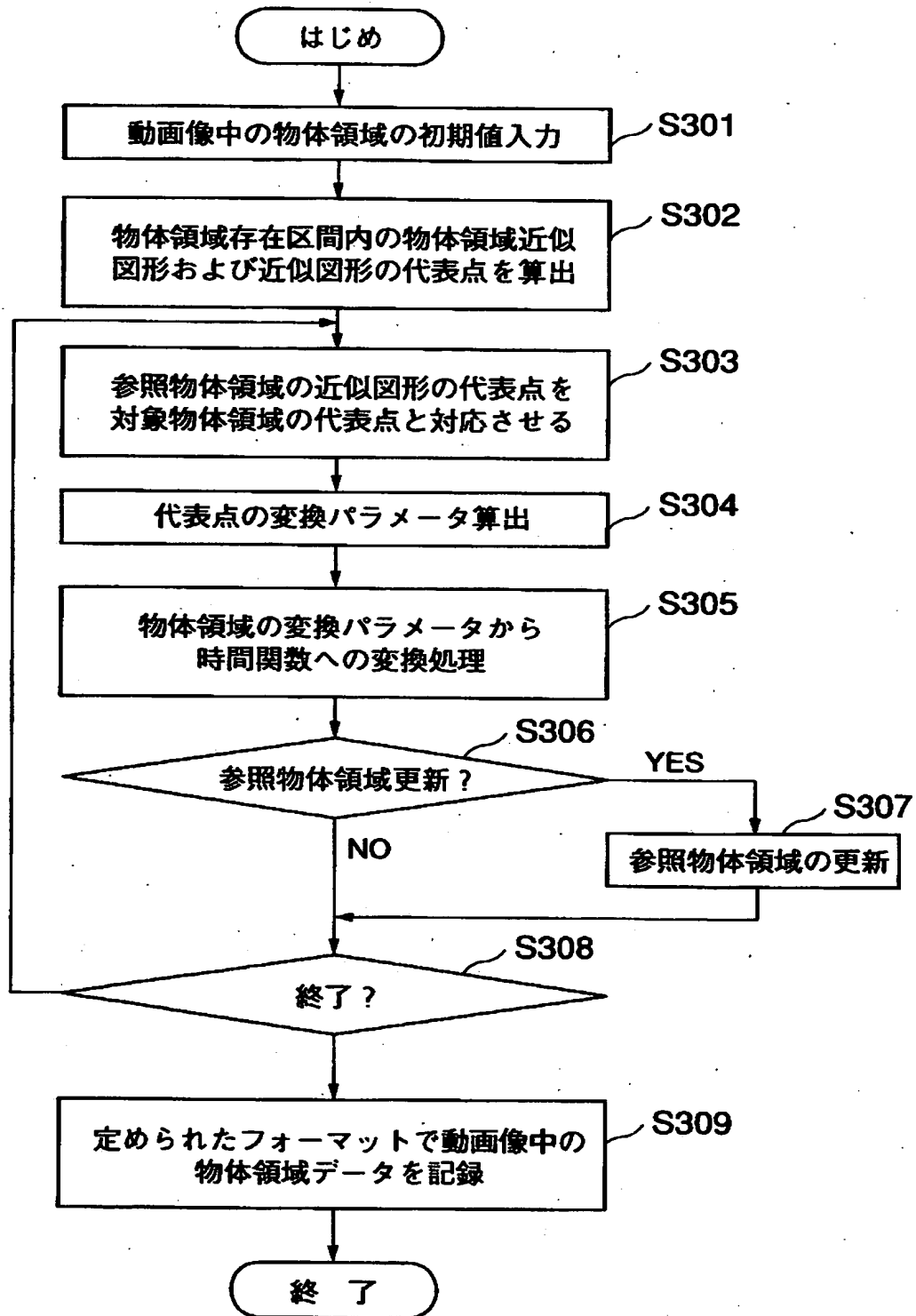
【図 7】



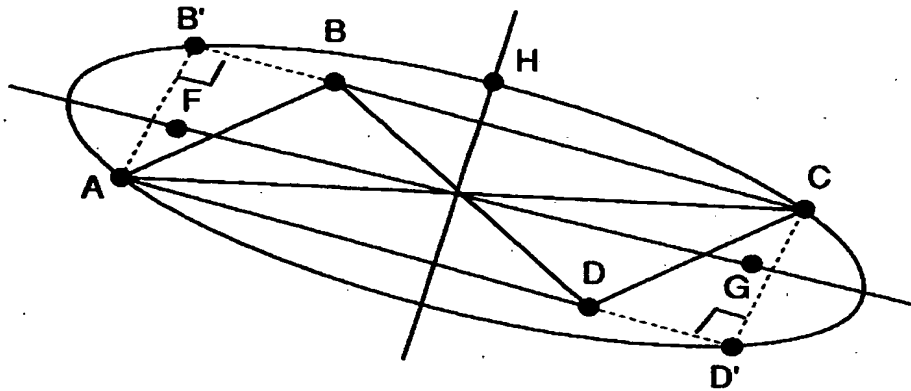
【図 8】



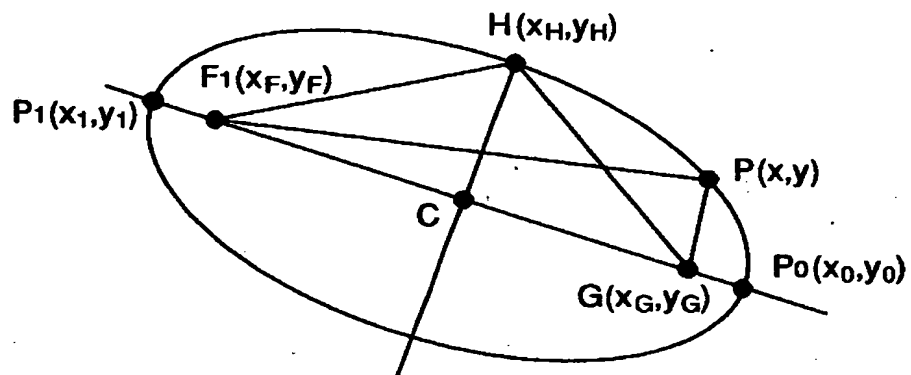
【図 9】



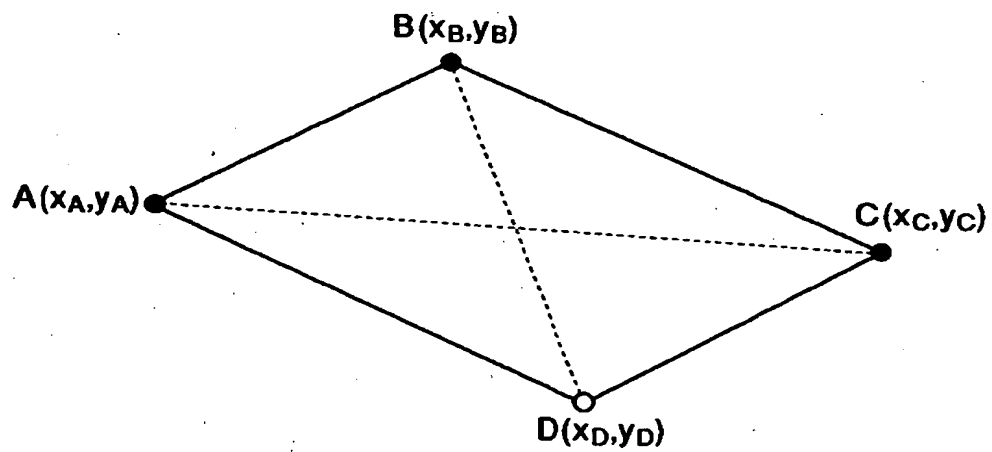
【図 1 0】



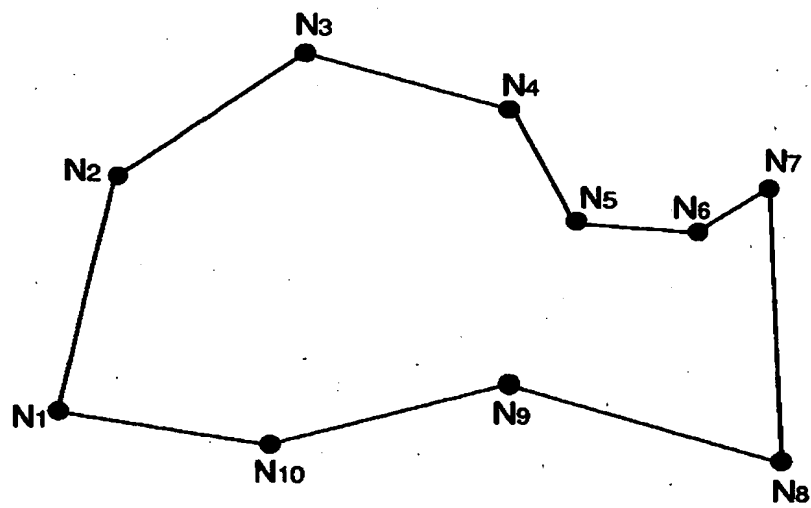
【図 1 1】



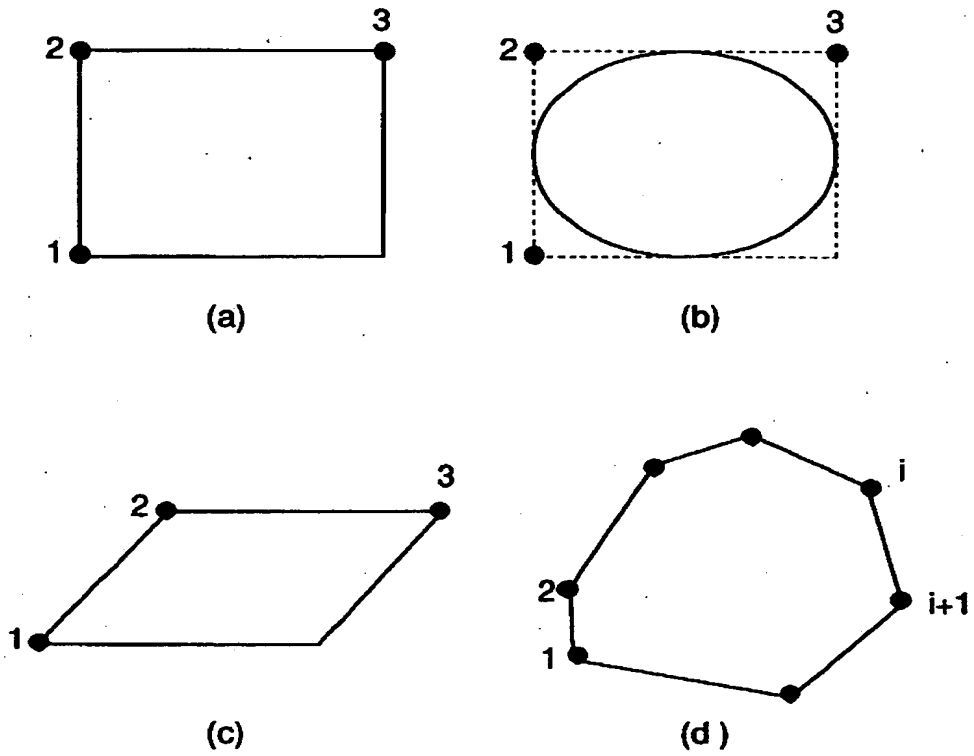
【図 1 2】



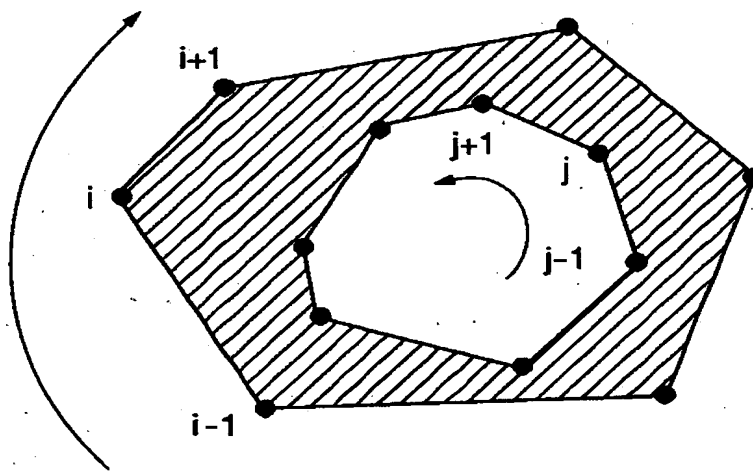
【図 1 3】



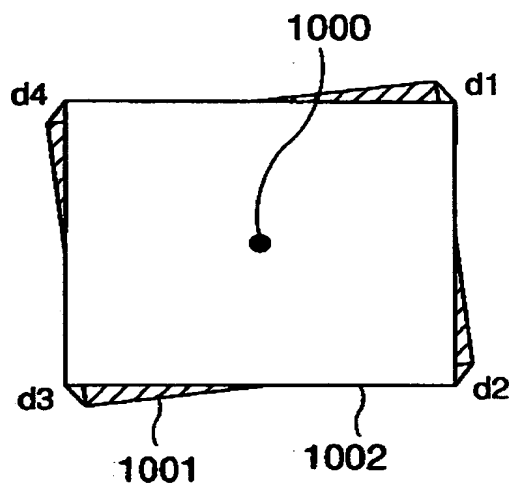
【図 1 4】



【図 1 5】



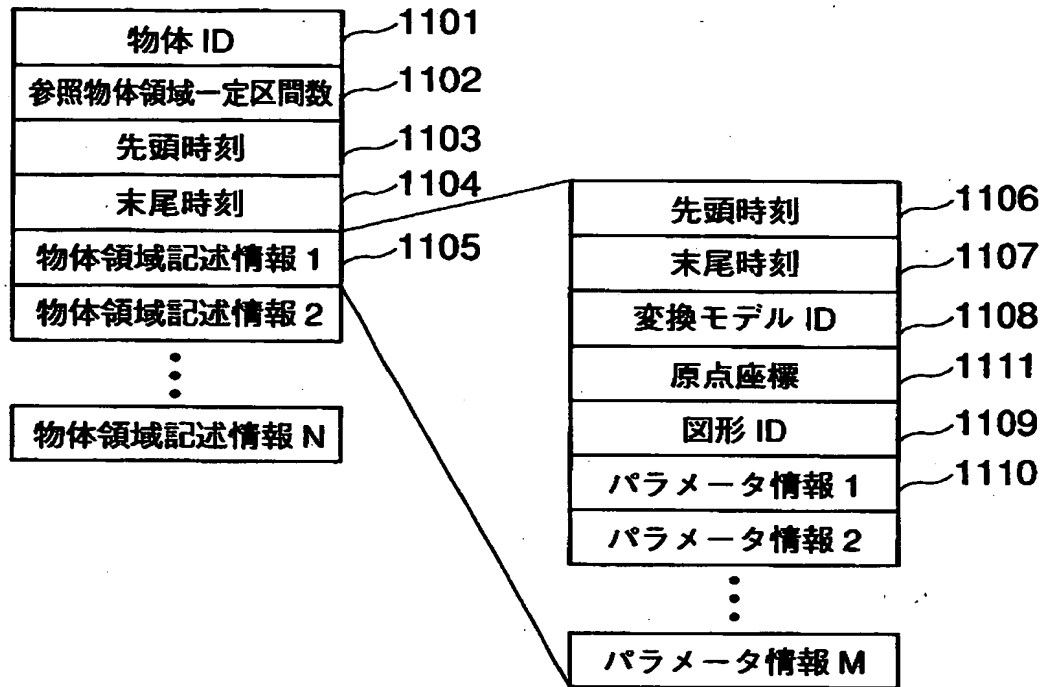
【図 1 6】



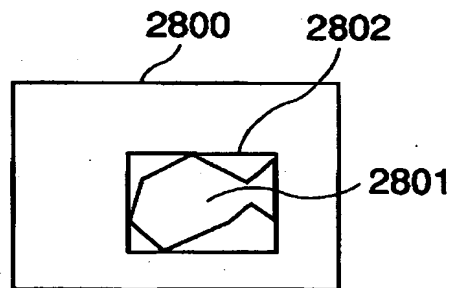
【図 1 7】

動きモデル 近似図形	拡大 縮小	回転	平行 移動	4-parameter 変換	affine 変換	射影 変換	parabolic 変換
矩形・楕円 (傾きなし)	○	×	○	×	×	×	×
矩形・楕円 (傾きあり)	○	○	○	○	×	×	×
平行四辺形	○	○	○	○	○	×	×
多角形 (頂点数>11)	○	○	○	○	○	○	○

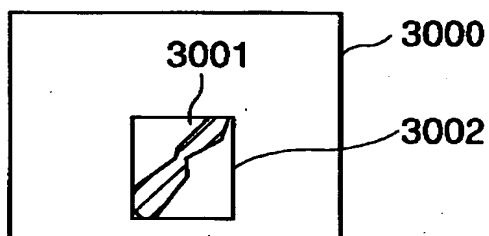
【図 1 8】



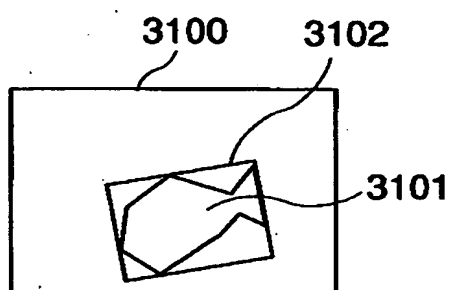
【図 1 9】



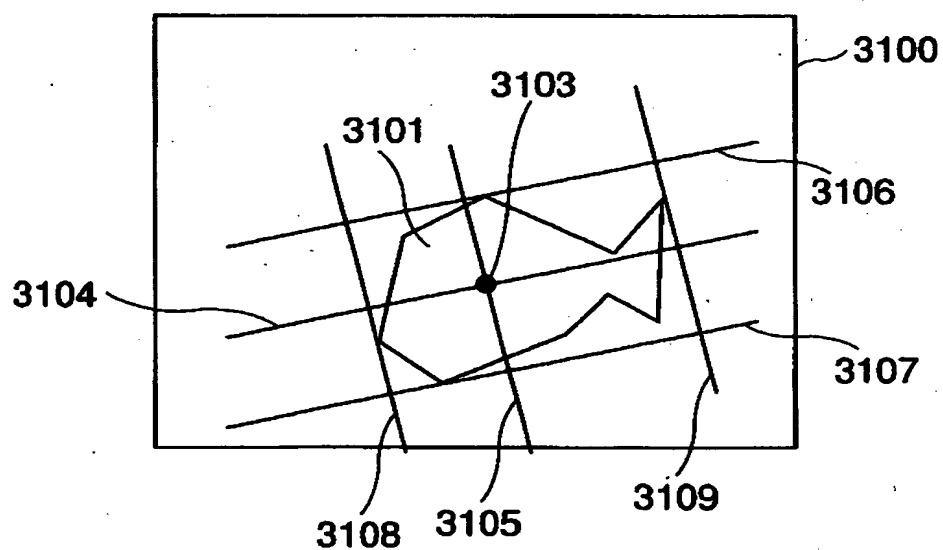
【图 2 0】



【图 2 1】

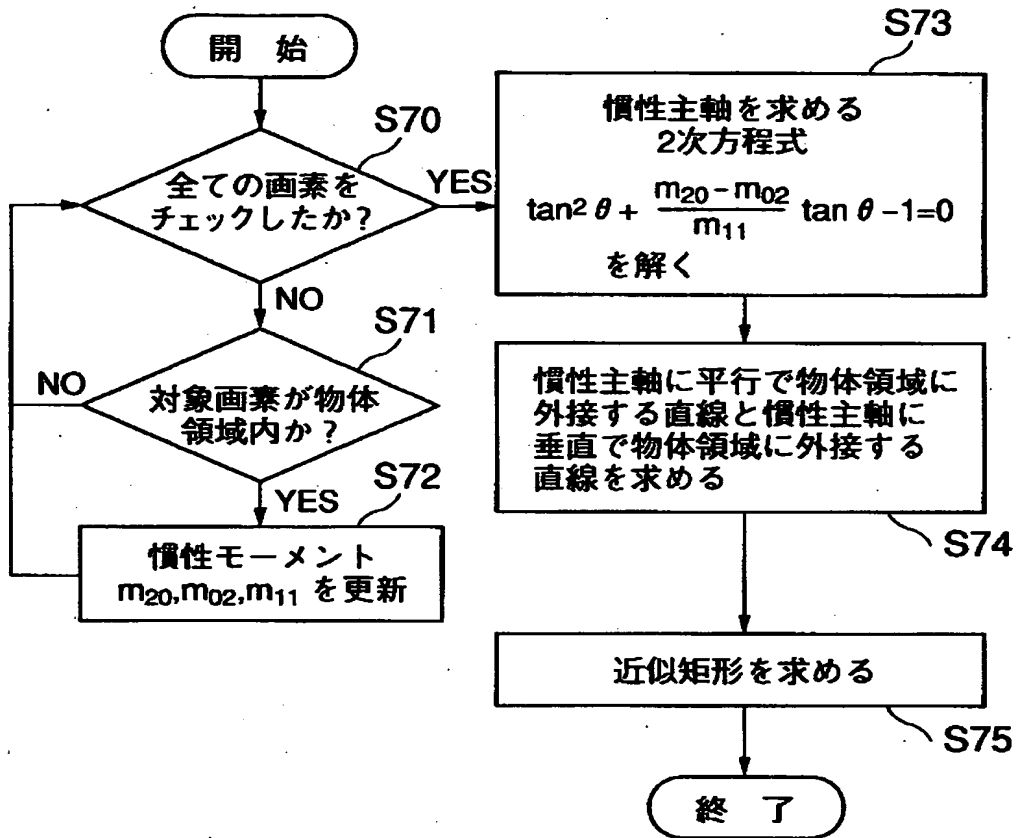


(a)

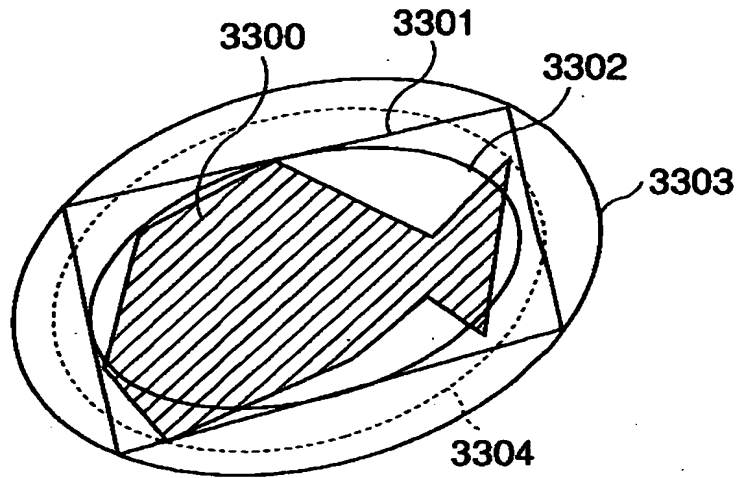


(b)

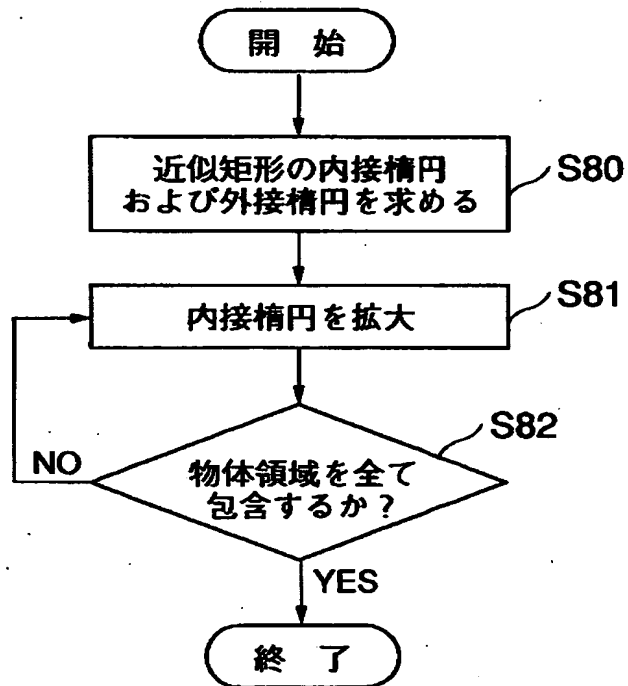
【図 2 2】



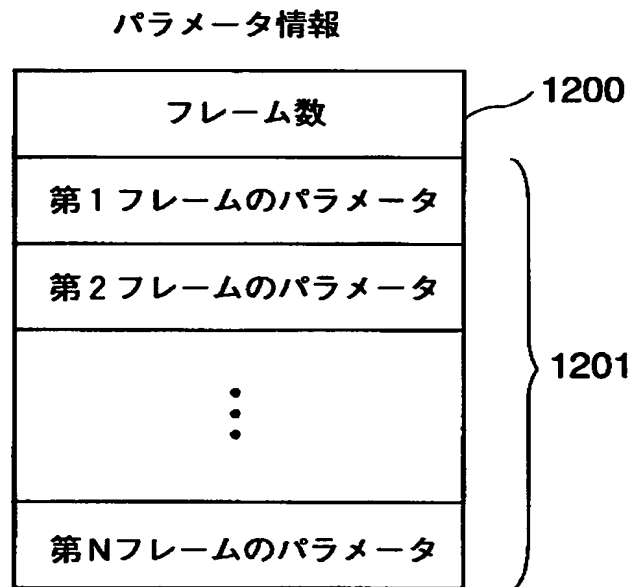
【図 2 3】



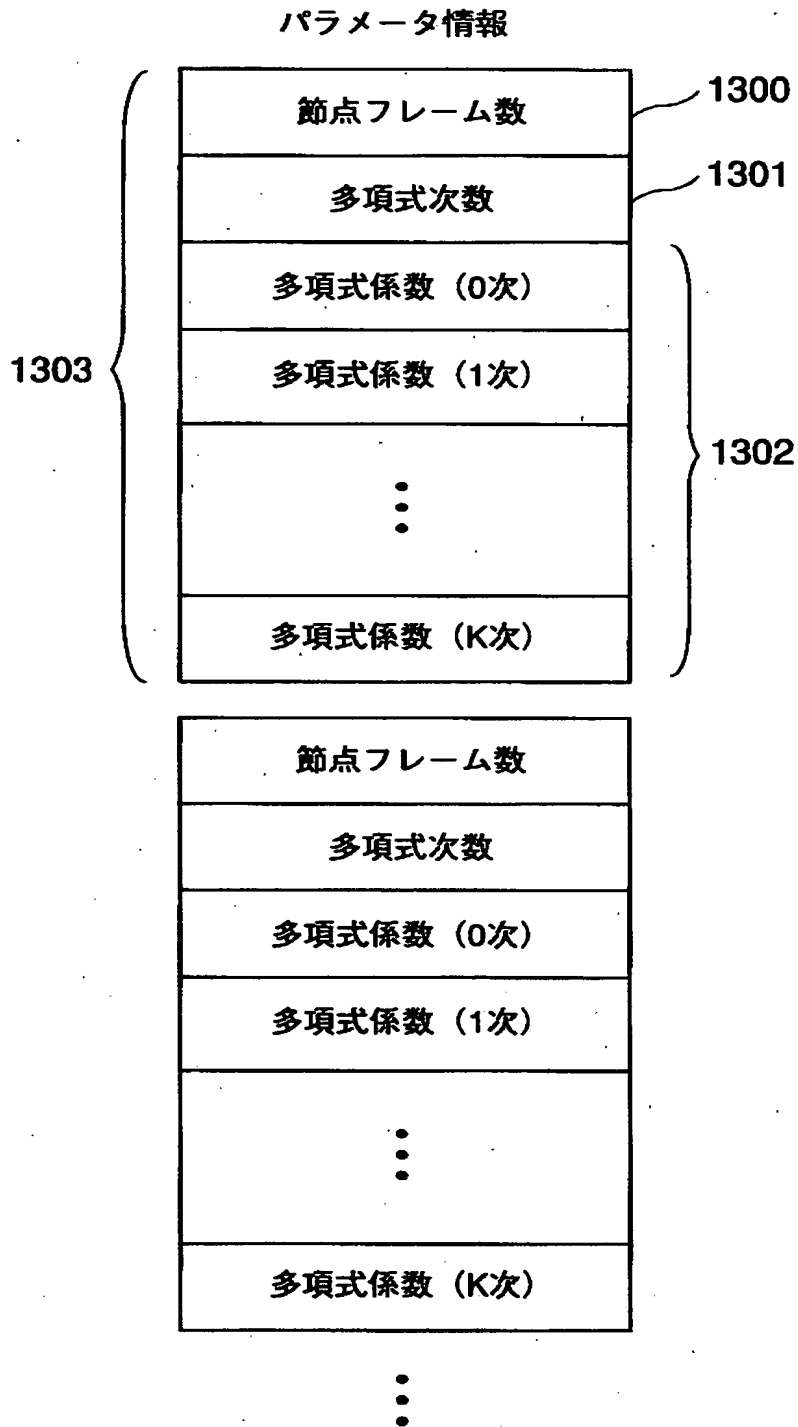
【図 2 4】



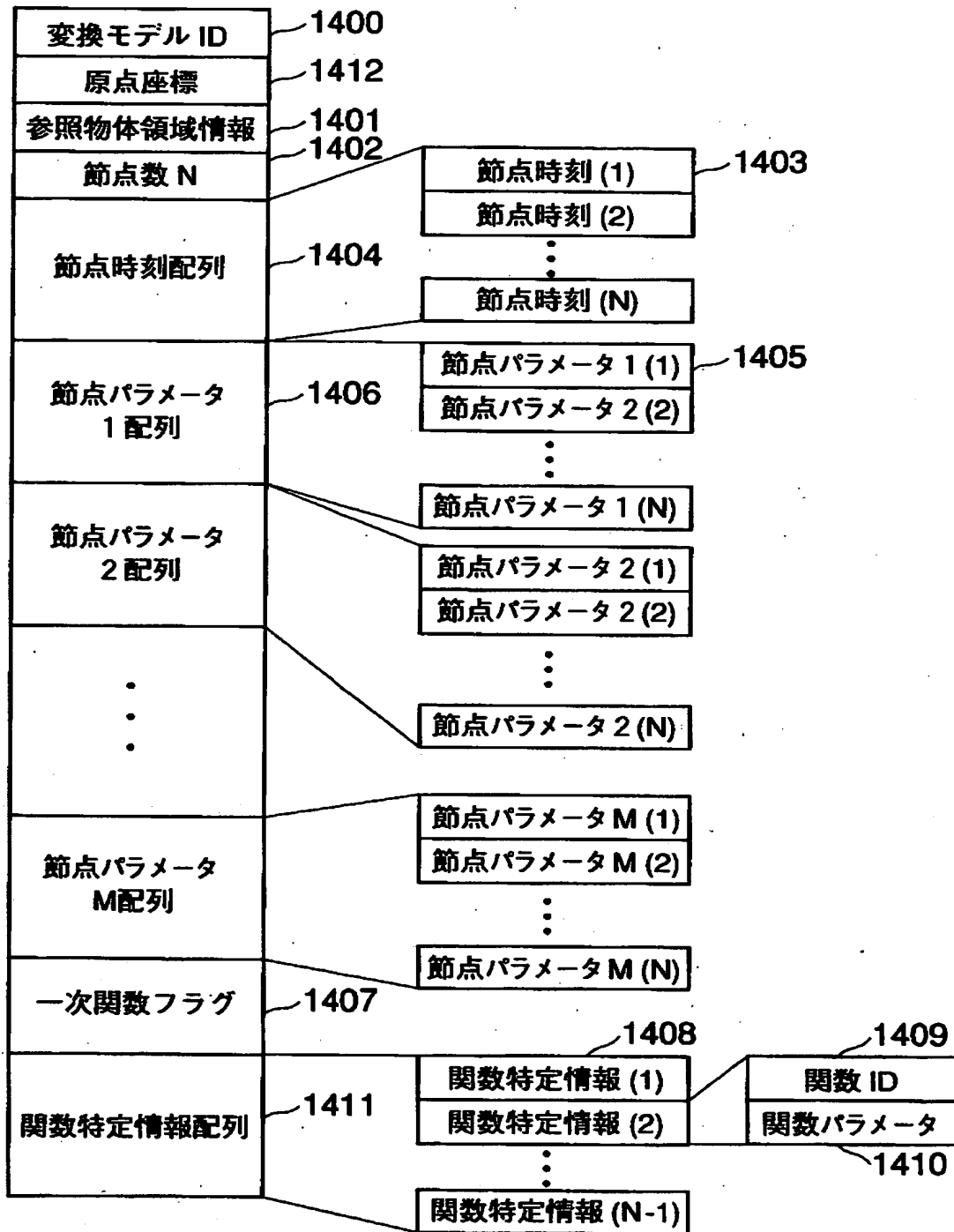
【図 2 5】



【図 2 6】



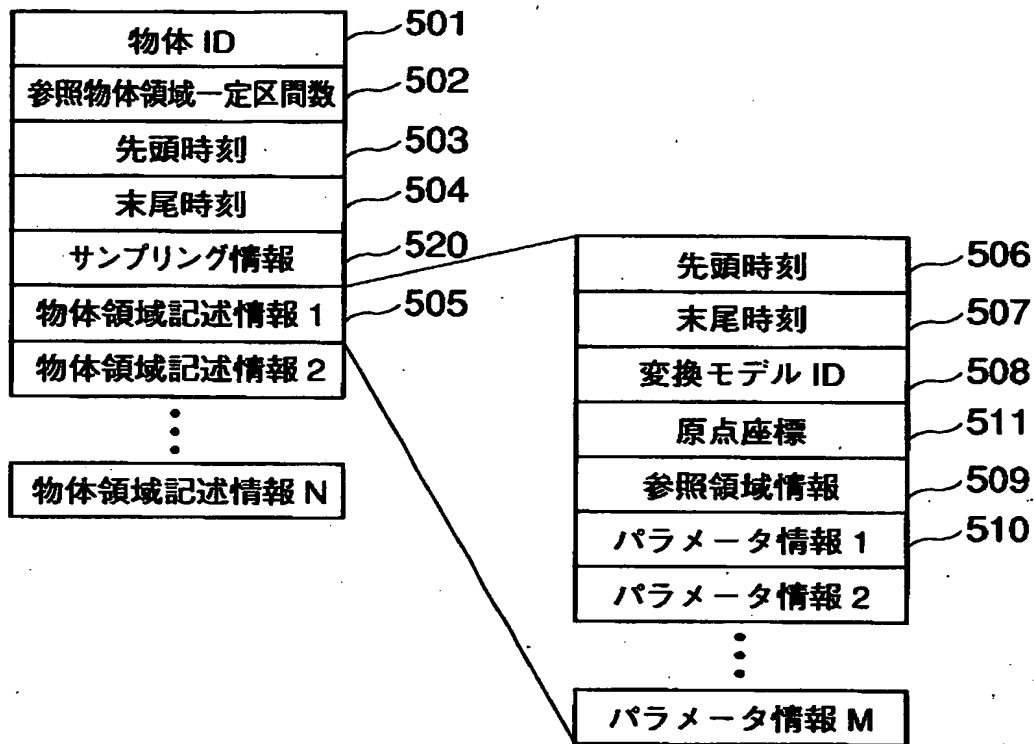
【図 2 7】



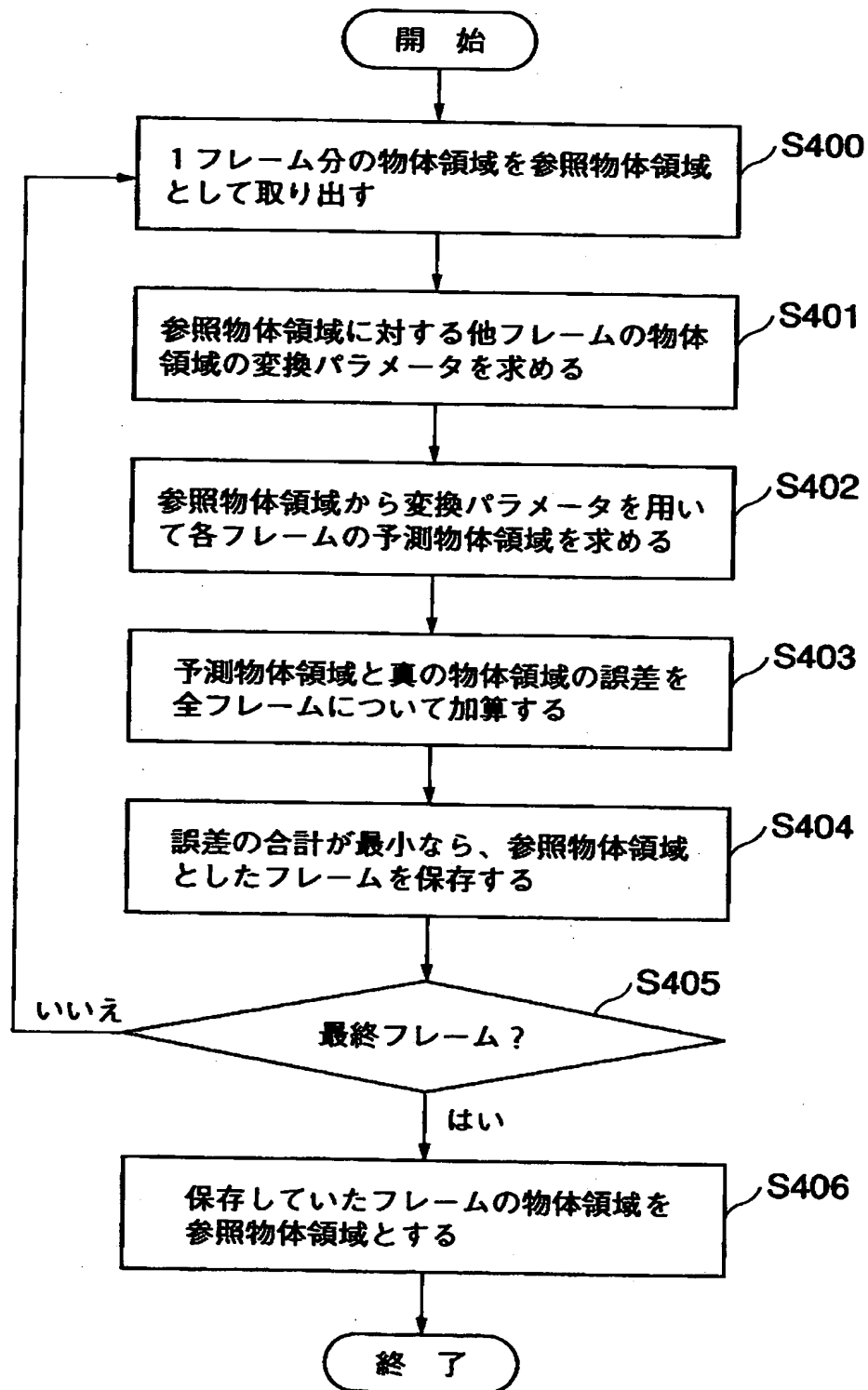
【図 2 8】

関数形式	関数パラメータ	制約条件
なし (関数は定義されない)	なし	なし
$f(t) = f_a + \nu_a(t - t_a)$	なし	$\nu_a = \frac{f_b - f_a}{t_b - t_a}$
$f(t) = f_a + \nu_a(t - t_a) + \frac{1}{2} a_a(t - t_a)^2$	a_a	$\nu_a = \frac{f_b - f_a}{t_b - t_a} - \frac{1}{2} a_a(t_b - t_a)$

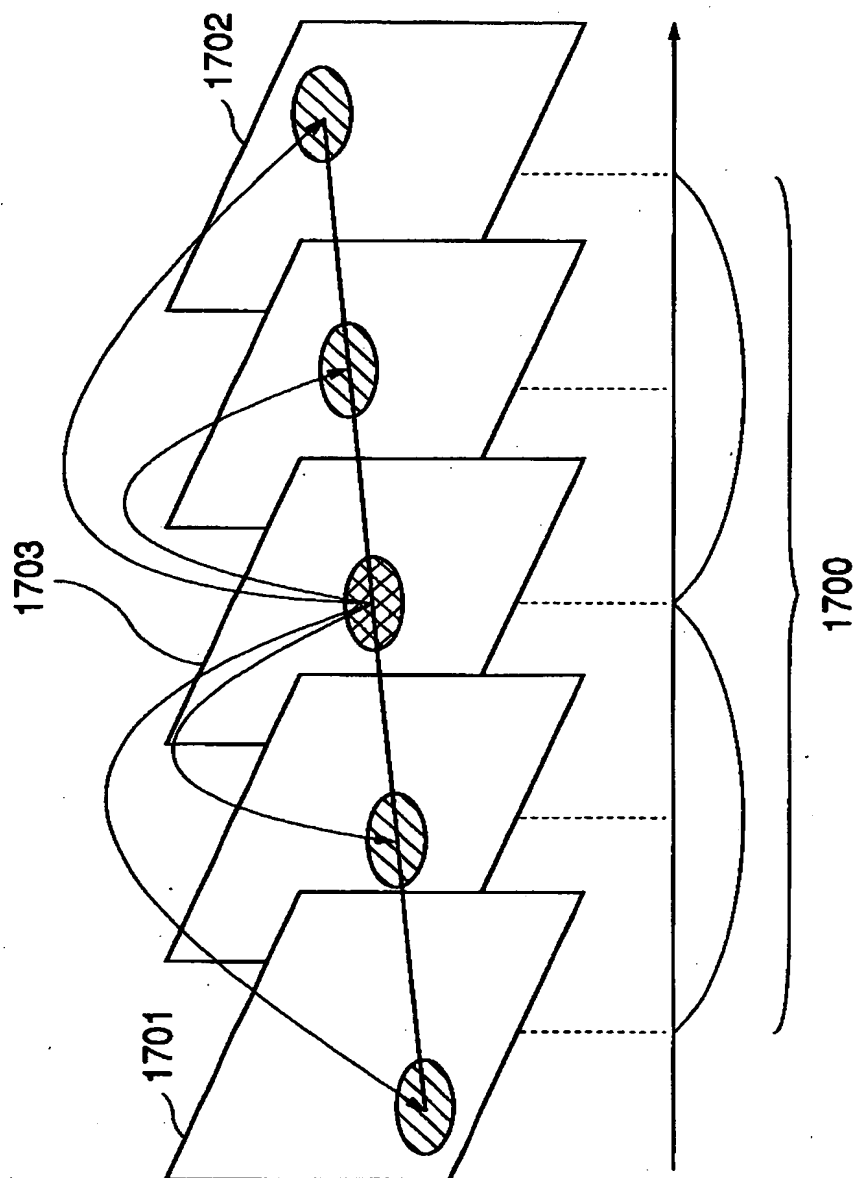
【図 2 9】



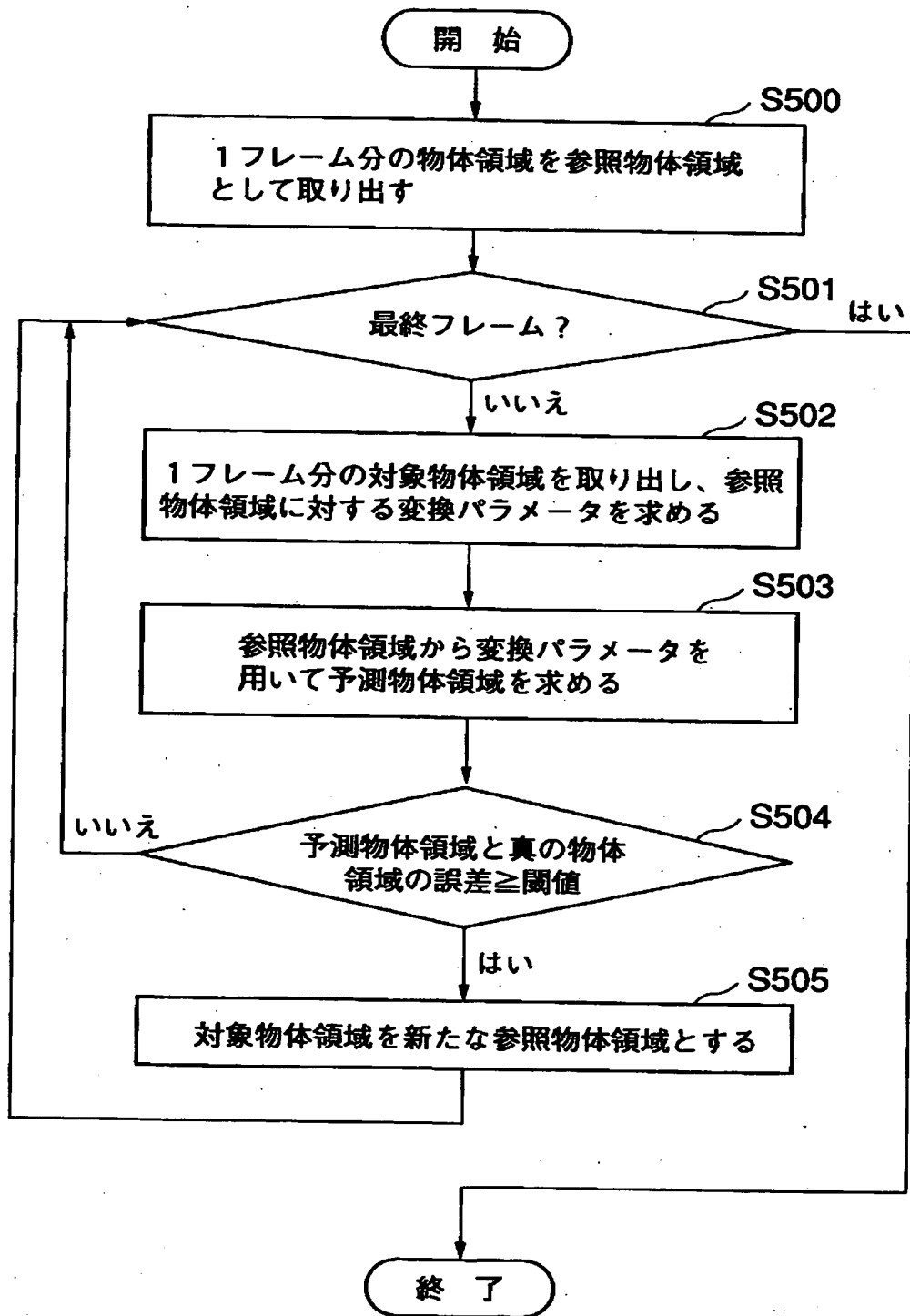
【図 3 0】



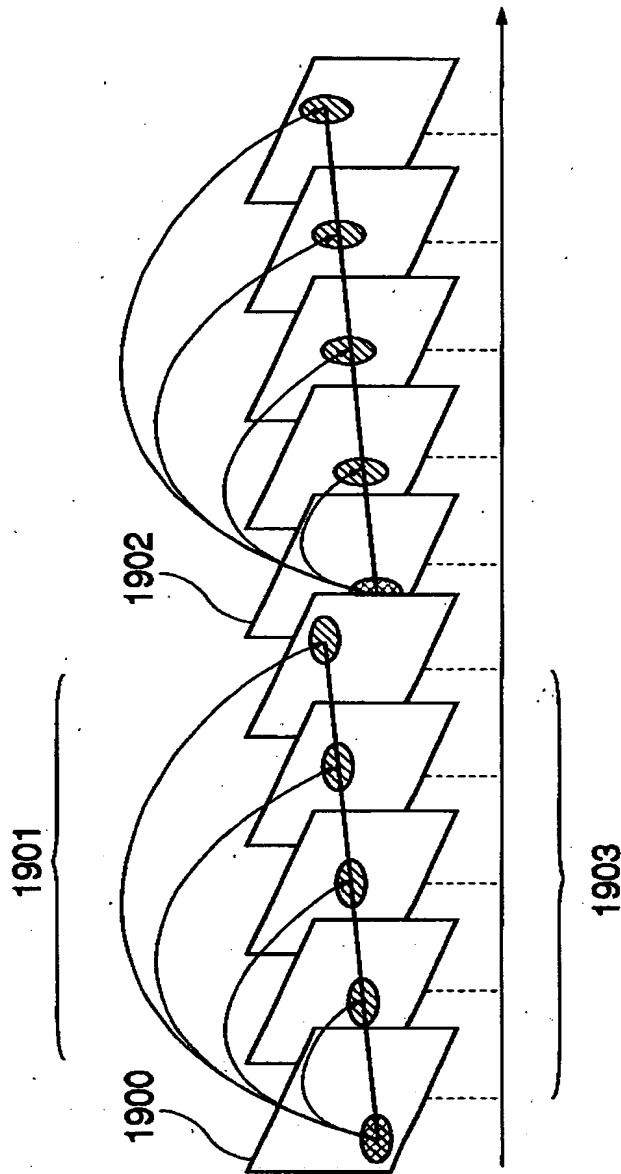
【図 3 1】



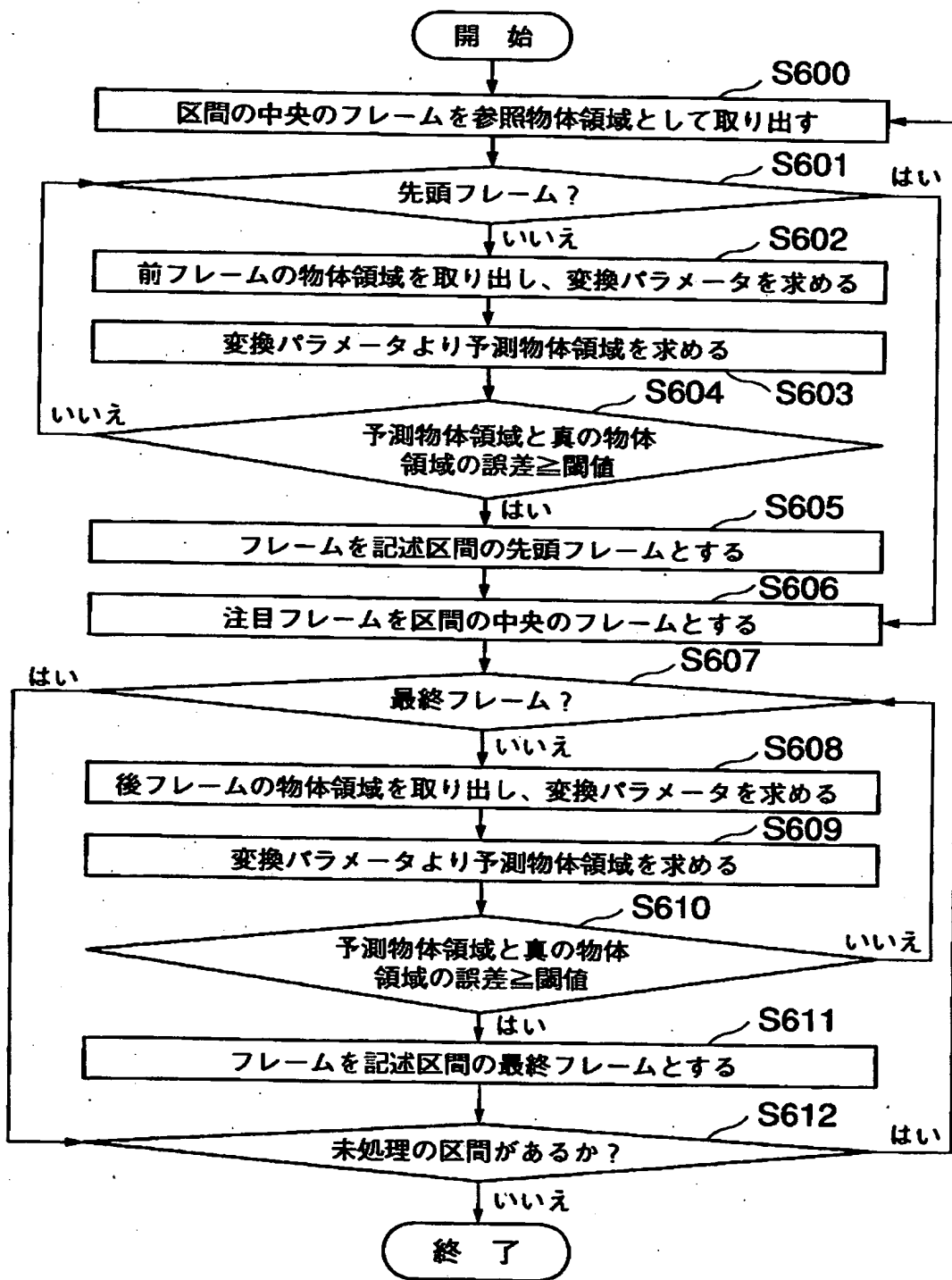
【図 3 2】



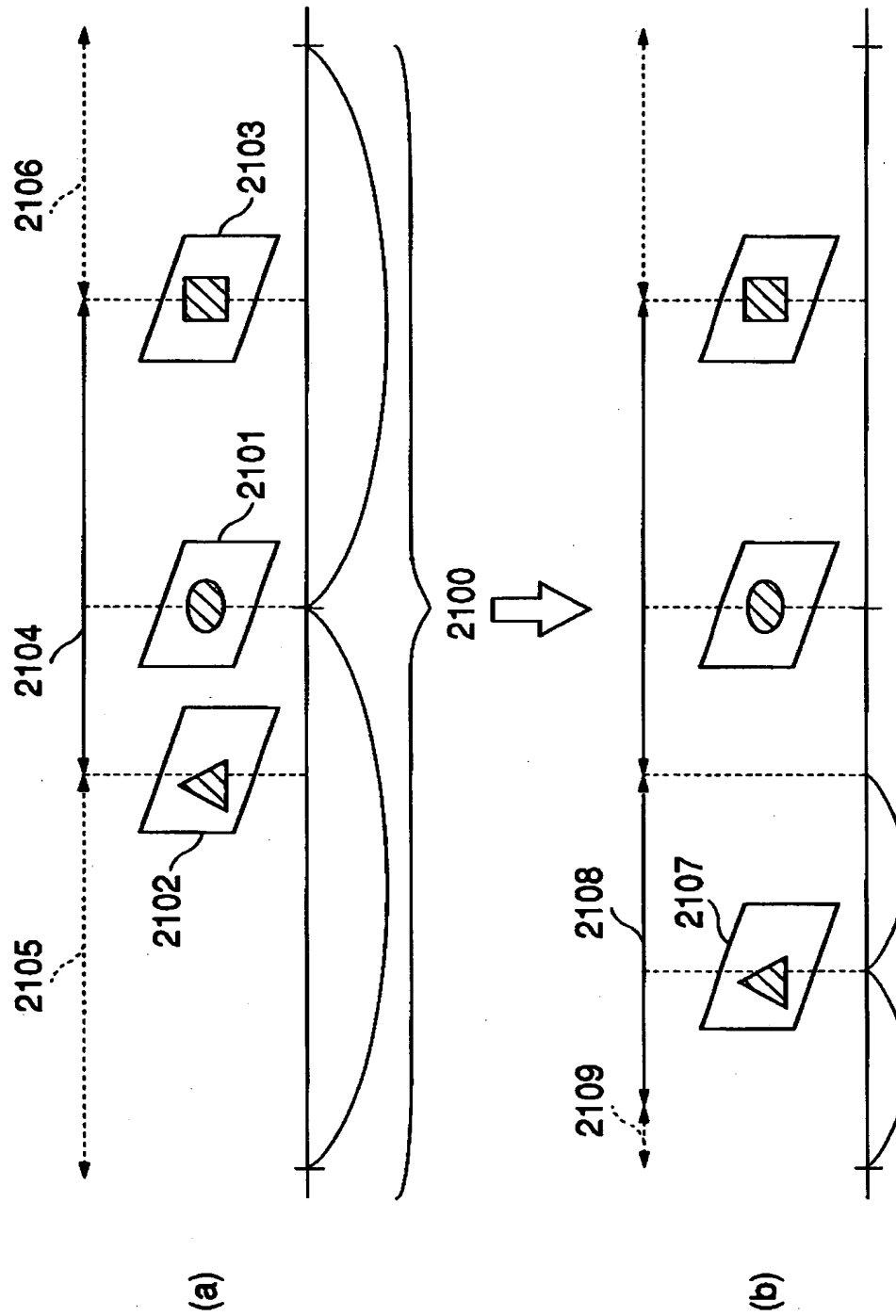
【図 3 3】



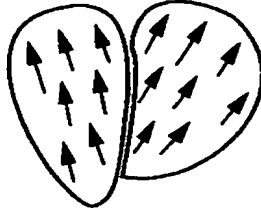
【図 3 4】



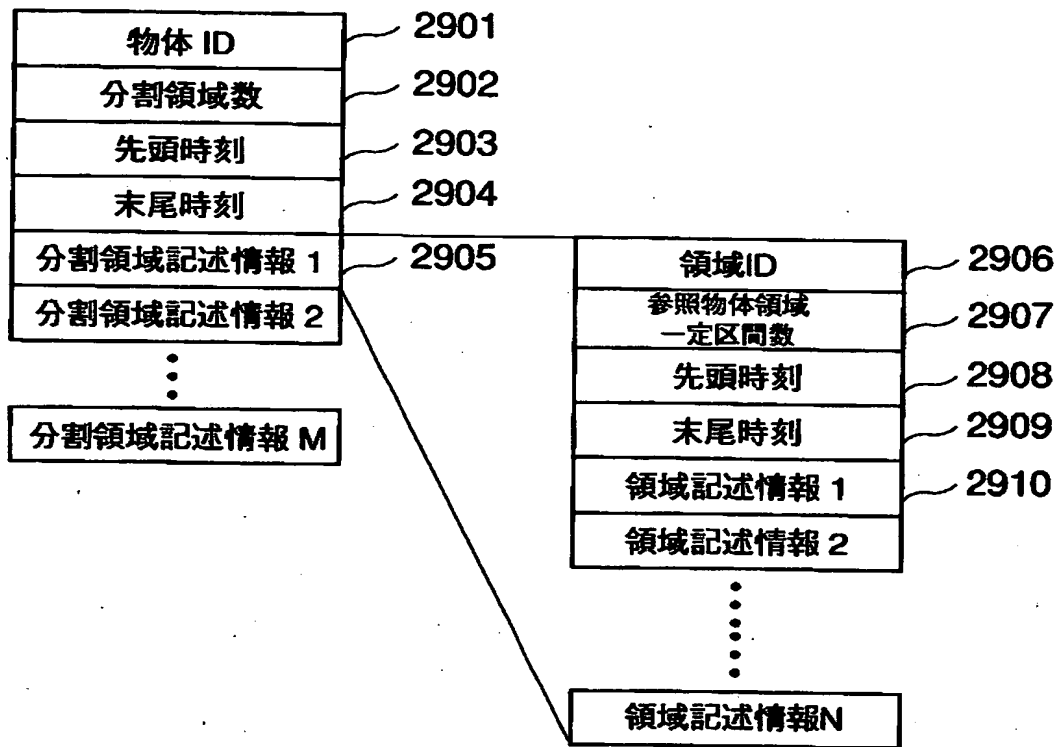
【図 3 5】



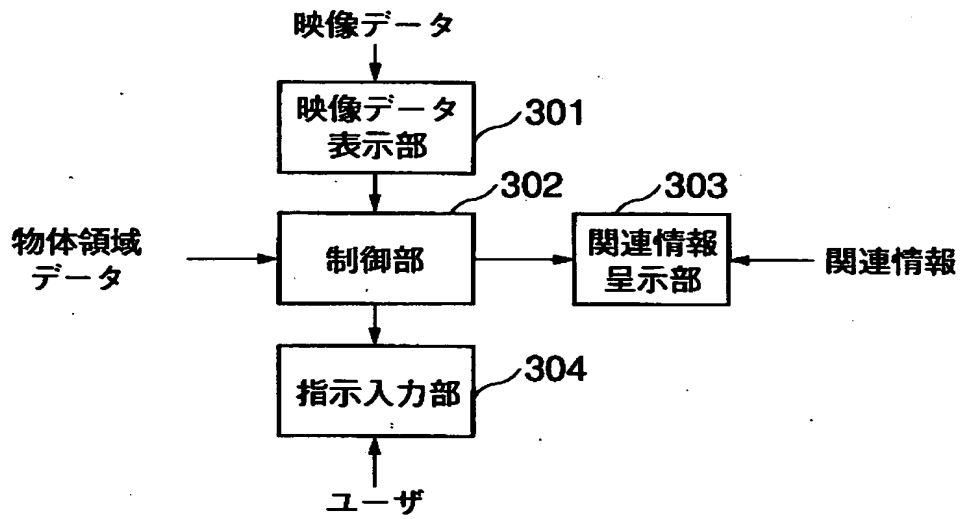
【図 3 6】



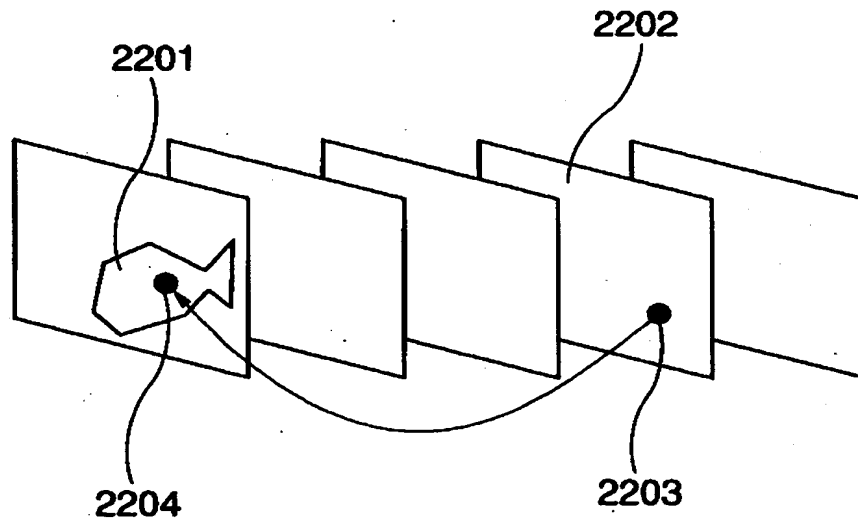
【図 3 7】



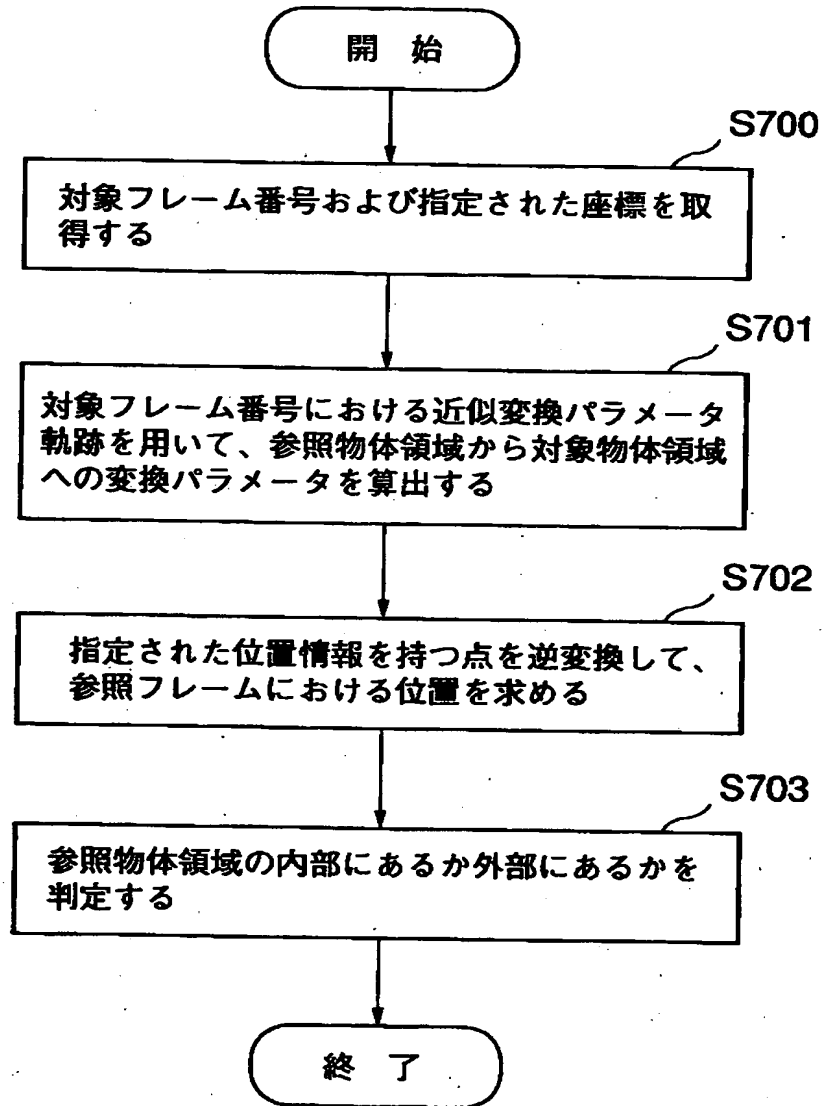
【図 3 8】



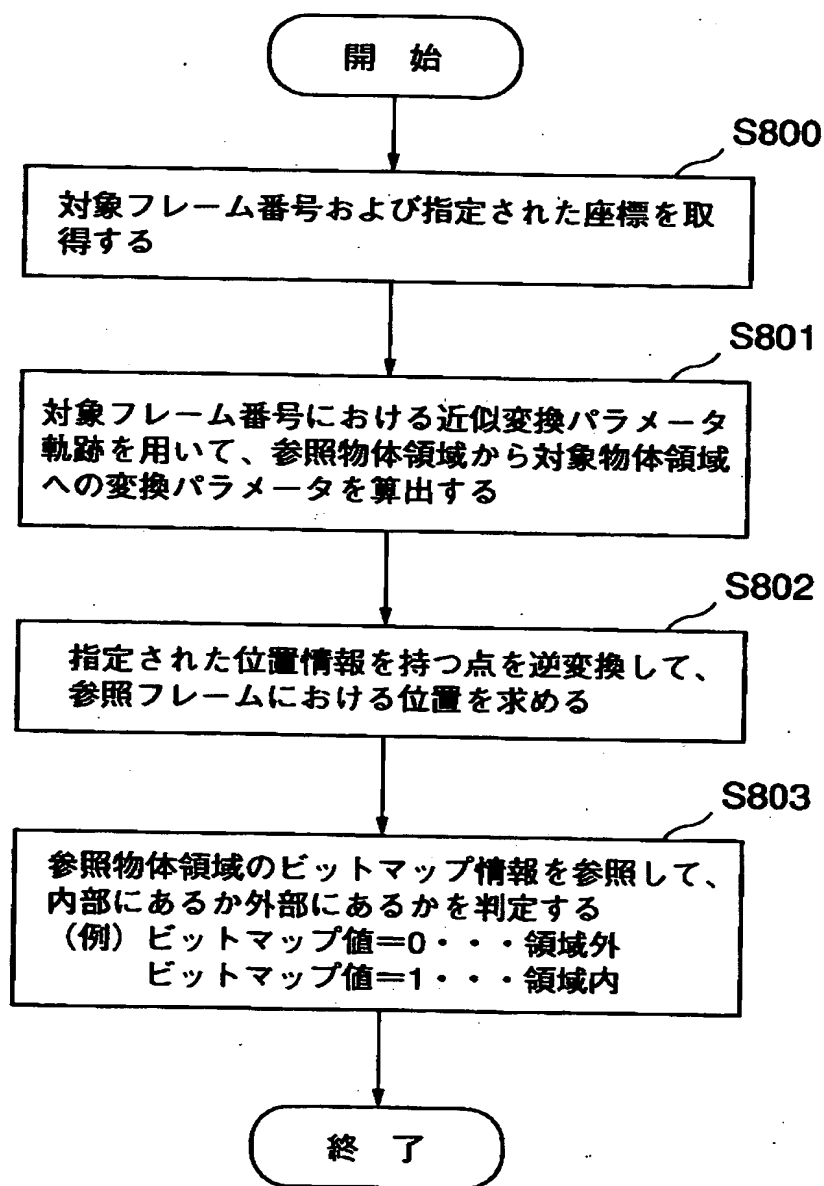
【図 3 9】



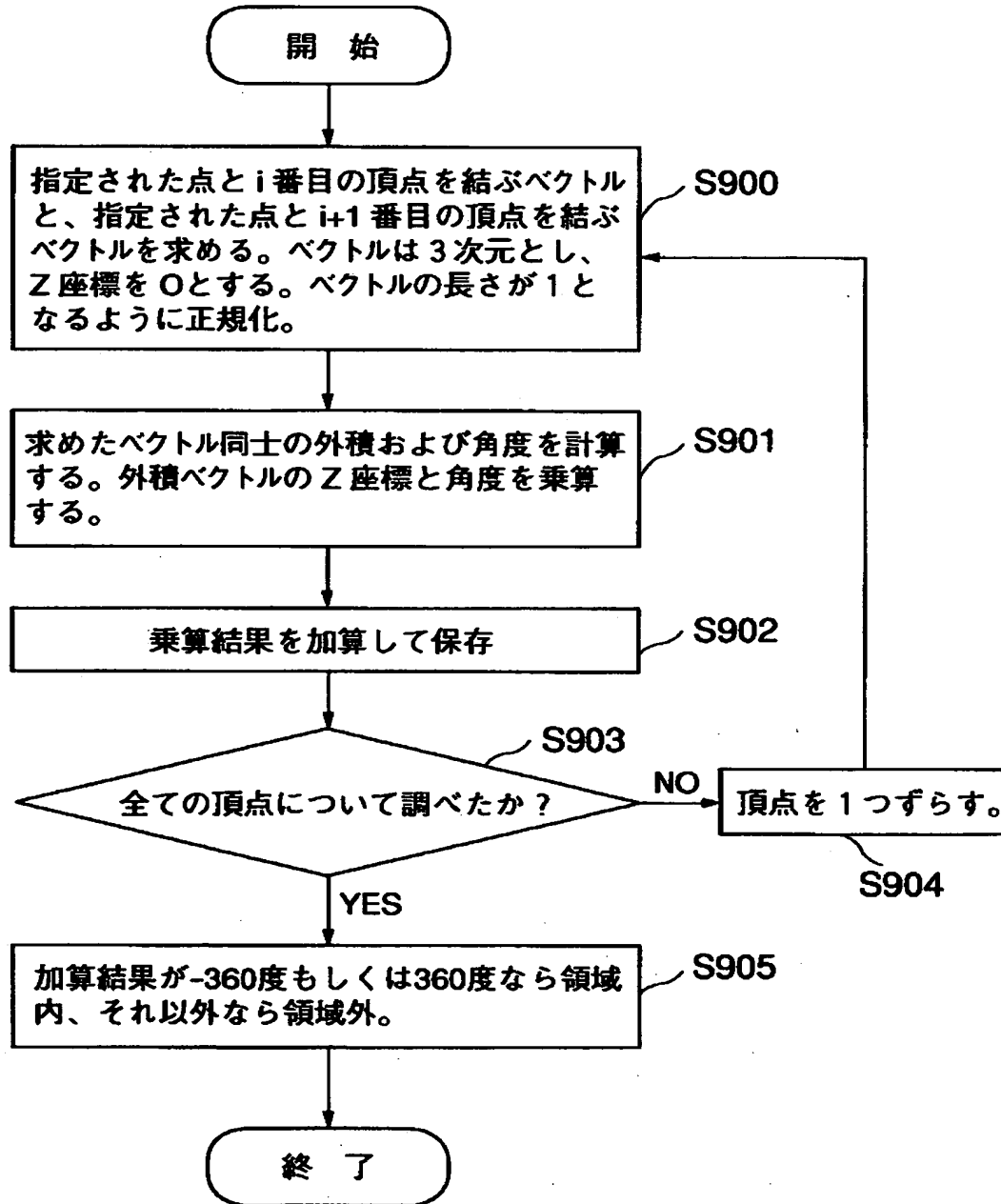
【図 4 0】



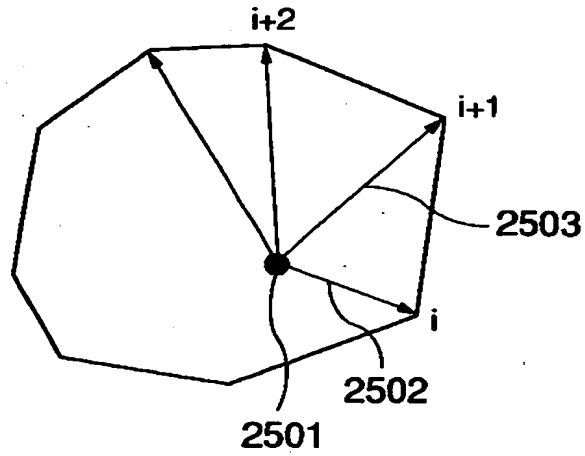
【図 4 1】



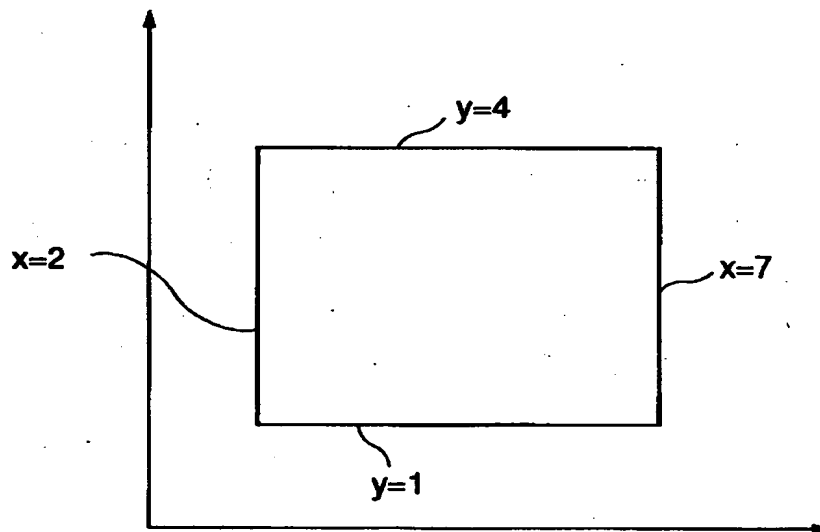
【図 4 2】



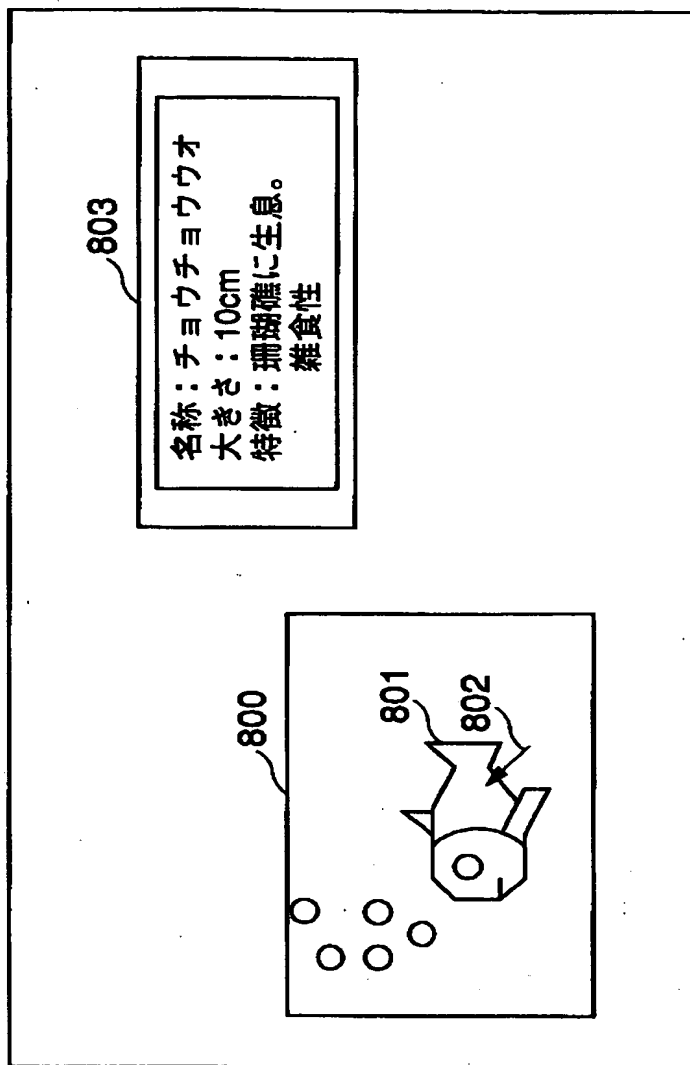
【図 4 3】



【図 4 4】



【図 4 5】



【書類名】 要約書

【要約】

【課題】 映像データ中の所望の物体の領域を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にする物体領域情報記述方法を提供すること。

【解決手段】 映像データ（201～203）中における物体領域（204～205）を表す物体領域データを記述するための物体領域情報記述方法であって、対象物体領域（205，206）を、参照物体領域（204）から当該対象物体領域（205，206）への領域変換を示す変換パラメータで表現し、変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似し、変換パラメータの各々についての、所定の近似関数を特定する近似関数パラメータと、該変換パラメータのもととなった参照物体領域（204）とを用いて、物体の領域に関する情報を記述する。

【選択図】 図3

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日	1990年 8月22日
[変更理由]	新規登録
住 所	神奈川県川崎市幸区堀川町72番地
氏 名	株式会社東芝